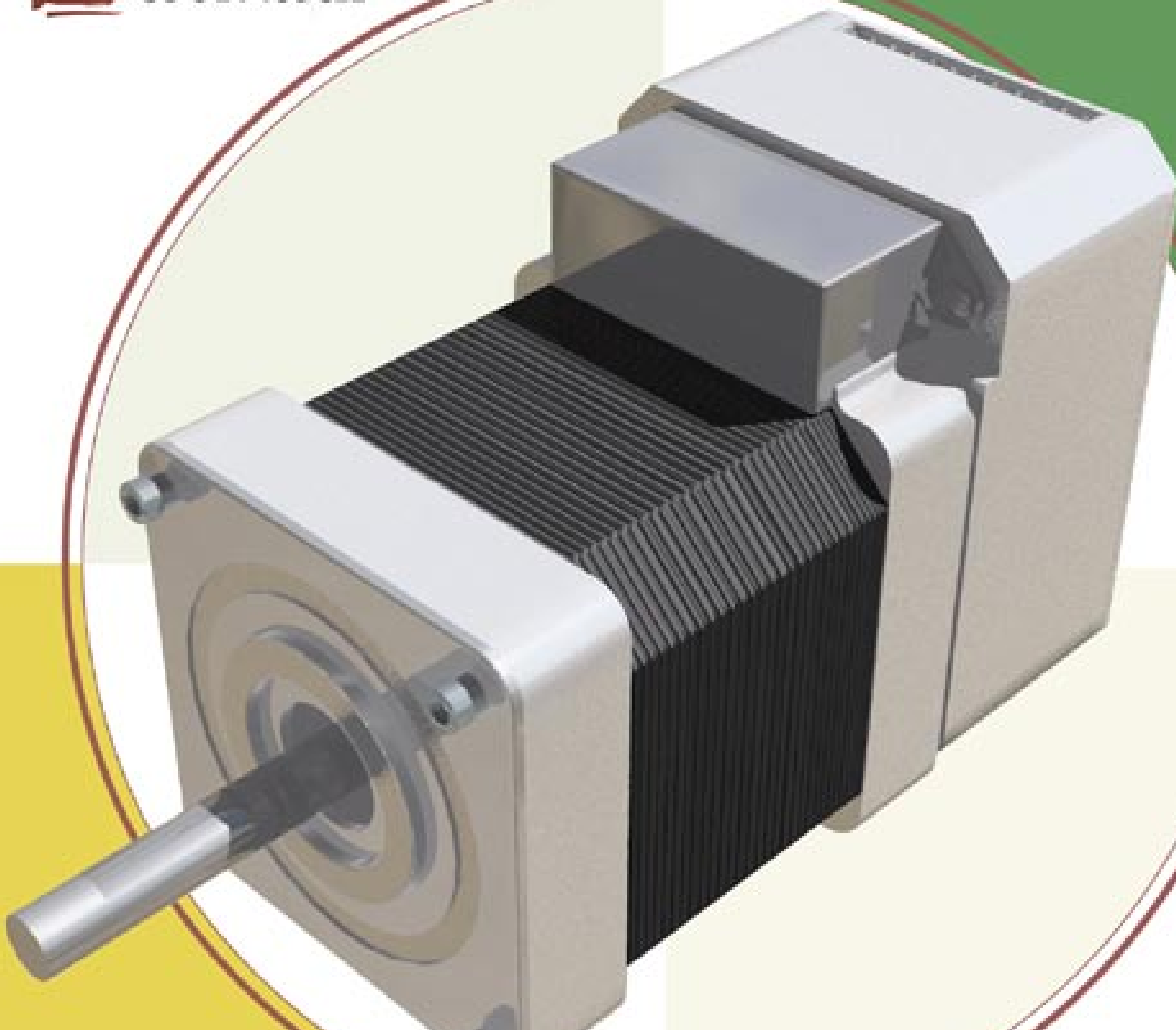
COOL MUSCLE

Integrated Servo System



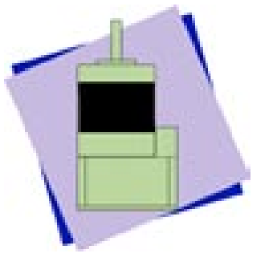
USER'S GUIDE

www.coolmuscle.com
EM-060324
For V2 Firmware

Copyright

Printed in Canada.

□□



PREFACE TO THE USER MANUAL

Welcome to the Cool Muscle integrated motor. The Cool Muscle User Manual provides you with hands-on instructions to help you set up your hardware, set parameters, create or edit command programs and make the most of the Cool Muscle motors.

Cool Muscle

The Cool Muscle is a fully integrated closed loop vector drive servo system. With an intelligent driver, a 32-bit RISC CPU, a magnetic encoder and power management built onto the motor, the Cool Muscle excels in performance, size and cost.

Supporting a wide variety of communication interfaces including USB, RS-232, Pulse (CW/CCW and Step/Direction) and Variable Voltage, the Cool Muscle can be used not only for new products but also for replacing or upgrading the current systems.

The Cool Muscle's high resolution encoder gives you the exceptional resolution of 50,000pulses/rotation. Combined with vector drive control, the Cool Muscle is an incredibly smooth and quiet motor.

Using CML (Cool Muscle Language), you can easily set parameters and create motion programs that can be downloaded directly onto the motor. (C type motors only)

Disclaimer

Before operating the Cool Muscle, it is very important that you read this User Manual thoroughly. The Cool Muscle can cause bodily injury and/or equipment damage if it is misused. Proper safety means and measures should be provided to prevent any misuse and/or improper operations. The user assumes all liabili-

ty for its use. The Cool Muscle shall not be used for mission critical applications without explicit written permission from Muscle Corporation.

Do not impact the motor.
Impacting the motor may cause the driver case and the sensor mounted on the back of the motor to move, resulting in alignment errors.

Do not rotate the shaft of the motor when it is not powered on.
Rotating the shaft over 400RPM when the Cool Muscle is not powered on may cause regenerated voltage within the motor. This regenerated voltage may damage the driver board.

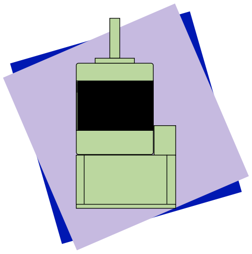
Use Regulated power supply
Please make sure to use a regulated DC24V power supply. Sudden spikes from non-regulated power supplies may cause damage to the motor.

Do not use the motor in wet conditions
Standard Cool Muscle motors are not environmentally sealed. Using the motor in wet conditions may damage the motor.

Use the motor within its specification
Please refer to the spec section of this manual and ensure that the motor is used within these specifications.

Are your cables too long?
The recommended length of motor cable is 3m and the power cable is 2m. If longer cables are required, please follow the instructions in this manual.

Do not connect or disconnect the motor connector when the motor is powered on. This may damage the motor.



SECTION 1

SET UP PROCESS

This section takes you through a typical Cool Muscle set up process. Since this User Manual is organized to follow the set up process described in this section, you can always refer back to this section.

Refer to the sections applicable to your Motor Type

- P Pulse Type
- V Analog Type
- C Computer Type

Step 1

When you receive the Cool Muscle

P V C

When you receive a Cool Muscle package, make sure that you have all the components you need or ordered. Connect the motor to your PC and make sure that there is proper communication between the motor and your PC.

See Section 2

Step 2

Connect Cool Muscle Cables

P V C

See how the Cool Muscle is connected using supplied cables. Learn about each component included in a standard Cool Muscle package.

See Section 3

Step 3

Learn I/O settings

P V C

Learn how to control I/O ports and set funtions for each I/O port.

See Section 4

Step 4

Lean about CML

P V C

Learn how CML (Cool Muscle Language) is structured.

See Section 5,6

Step 5

Set Parameters

P V C

Set parameters to make the Cool Muscle suit your specification. Learn how to set the parameters. This section also lists all the parameters available.

See Section 7

Step 6

Create/Edit/Execute Motion Programs

C

Learn all Programe mode CML commands and how to create motion programs with parameters, I/Os and commands. Useful basic examples are described.

See Section 8

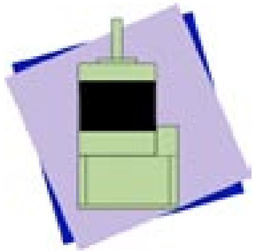
Step 7

Operate Cool Muscle with Dynamic Commands

C

Lean how to operate Cool Muscle using Dynamic CML commands.

See section 9



SECTION 2

WHEN YOU RECEIVE COOL MUSCLE

When you receive a Cool Muscle package please make sure that you have all the components you ordered. You may then proceed to the next step and connect the motor to your PC to make sure that you have proper communications.

Do you have...

The components that you need to operate the Cool Muscle are listed in the green area in the table below. The components listed in the pink area are optional components used for networking the Cool Muscles.

Components /Cool Muscle Type	Pulse Type	Analog Type	Computer Type
Cool Muscle Motor	O	O	O
Motor Cable	O	O	O
Y Cable	Option	Option	Option
24V Cool Muscle Power Supply	Option	Option	Option
CD*	O	O	O
Network Card (master set)	Option	Option	Option
Network Card (slave set)	Option	Option	Option
D-sub 9pin network cable (straight male-female)	Option	Option	Option

O=included with the motor

Option=optional component

*Intelligent USB comes with a power cable. You need a standard USB cable to connect the Hub to your PC.

Components

Motor Cable
The Cool Muscle comes with one motor cable. The standard length of the motor cable is 40cm. Longer cables are available from your local Muscle distributor.



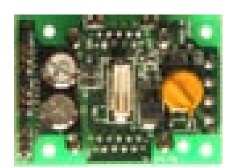
Y Cable
A Y cable is required to connect the Cool to a RS-232C port on a PC. This cable is required when downloading motor parameters, and motion programs onto the Cool Muscle, or when you wish to control a single motor via serial communication. A Y cable can be ordered from your Muscle distributor.

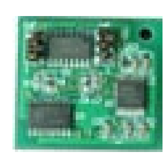


Power Supply
A regulated 24V (DC) power supply. Required power supplies may be available from your local Muscle representative.



Network Card and Serial Daughter Card
The Network Card allows you to daisy chain the Cool Muscles. A master motor requires a serial daugher board in addition to the network Card. The Network Card can also be used as a voltage clipper. D-Sub 9 pin cables (straight) are required to inter-connect the Cool Muscles and to connect a PC and a master motor.





Communication with the Cool Muscle

Connect the Cool Muscle to your PC to make sure that your PC can communicate with the motor. Please follow the steps below. A trouble shooting section is included on the next page.

1. Connect the Y cable to the Cool Muscle, your PC and a 24VDC power supply. (refer to Section 3 “Setting Up Cool Muscle”)
Make sure that the power is turned OFF.

2. Start up your PC

3. Start up Cool Works Lite. Cool Works Lite is available on the CD provided with the Cool Msucle, or can be downloaded from our websited at www. coolmuscle.com

4. Power the motor on. Motor information should appear. If motor information is displayed you are ready to talk to the motor!



5. Check the preloaded parameters by entering ?90 and press the Enter key. Parameters (starting with K25) should appear in the motor reply window. These are the preloaded parameters.

6. Turn off the motor and close Cool Works Lite

Trouble Shooting

Nothing appears when the motor is powered on.

Is the Cool Muscle Powered on? Make sure that the motor is not in motor free state. If it is turn the power off and then turn it on.

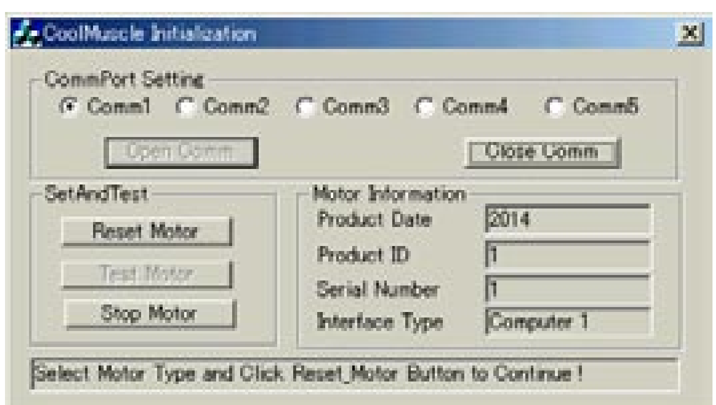
Do you have multiple terminal applications running in the background? Or do you have software for PDAs or other programs that occupy COM ports running? Make sure that these programs are disabled and not running in the back ground.

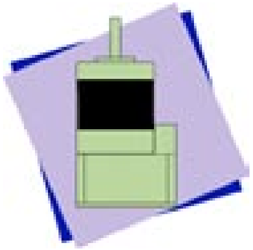
Nothing appears when ?90 is entered.

Parameters are not properly displayed. Turn the motor off and on. If the motor responds with its version information, communication is established.
You can load default parameters with the MotorReset software provided on the CD.

1. Start MotorReset software on the CD included with the Cool Muscle.

2. Click Reset Motor to load the default parameters. Then, click Test Motor to make sure that the Cool Muslce operates properly. Click Stop Motor to stop the Cool Muscle.





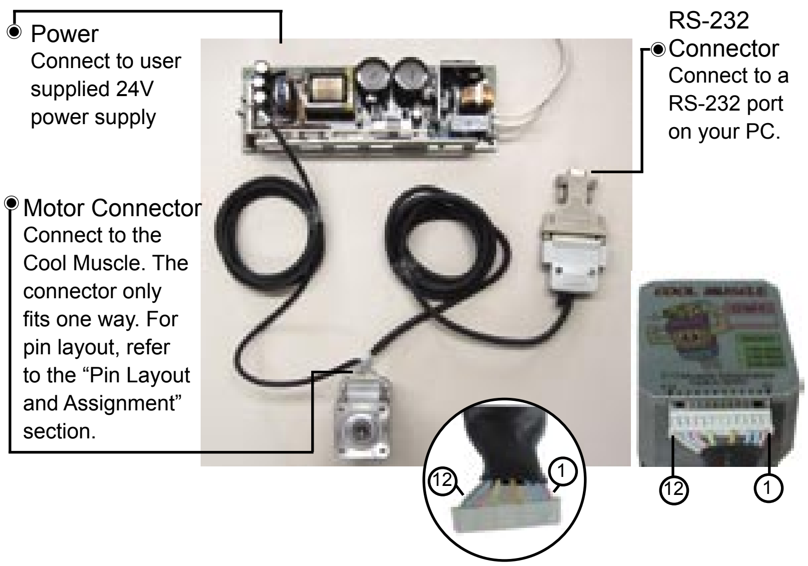
SECTION 3

SETTING UP COOL MUSCLE

How To Connect The Cool Muscle

Using Y cable

Do NOT connect or disconnect the motor while it is powered on.



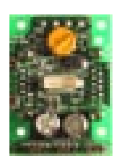
Using Network Cards

You can network multiple motors by using network cards.

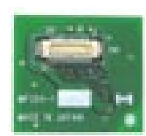
Instead of using the Y cable, you can use the network card and the interface card to connect the Cool Muscle to a PC. You need a D-Sub 9 pin straight cable

What Do I need to Create a Daisy Chain?

1. Network Card
2. Serial Interface Card
3. Motor Cables
4. D-Sub 9 pin Straight cables
5. DC+24V Power Supply
6. Power Cable



Network Card



Serial Interface Card

Jumpers

Before you connect the network and interface cards, you need to set the jumpers.

Network Card

Jumper 1&2

These jumpers are for switching the DC+24V line on the card.

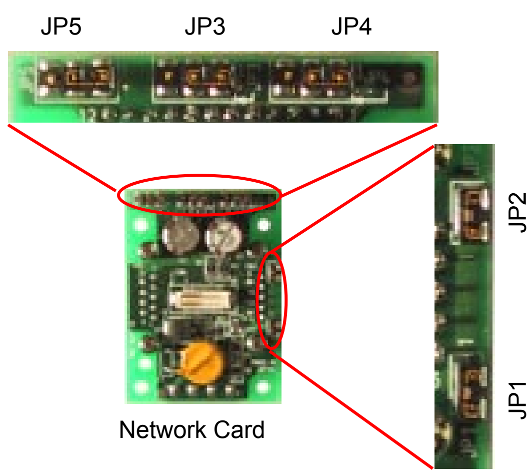
• Leave JP1&JP2 open when power is supplied through the terminal block on the card.

• Close JP1&JP2 when power is supplied via the serial cable.

Jumper 3,4&5

• Connect Pin 2 and 3 of each Jumper when a Serial Daughter Card is attached to the network card (Master Network Card - CM1DC1-MBS).

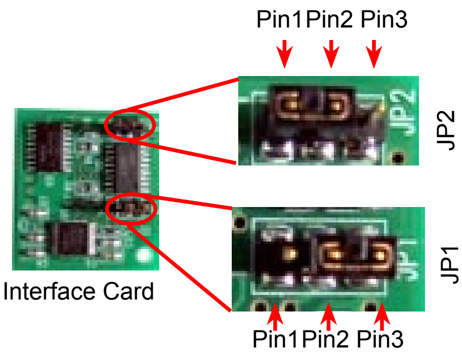
• Conect Pin 1 and 2 when a Serial Daughter Card is NOT attached to the network card. (Slave Network Card - CM1DC1-SBS)



Interface Card

JP1 and JP2 on the interface board allow you to switch between RS232 and RS485.

JP1 JP2	RS232 Connect Pin 1&2 Connect Pin 2&3
JP1 JP2	RS485 Connect Pin 2&3 Connect Pin 1&2



Interface Card

How to Connect the Network Card and the Interface Card

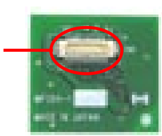
To daisy chain the Cool Muscles,you need network cards and an interface card. A master motor which will be connected to a host computer needs both the network card and the interface card. Slave motors need only the network card.

Master Motor

Attach the interface card to the network card.

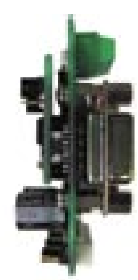


Connector



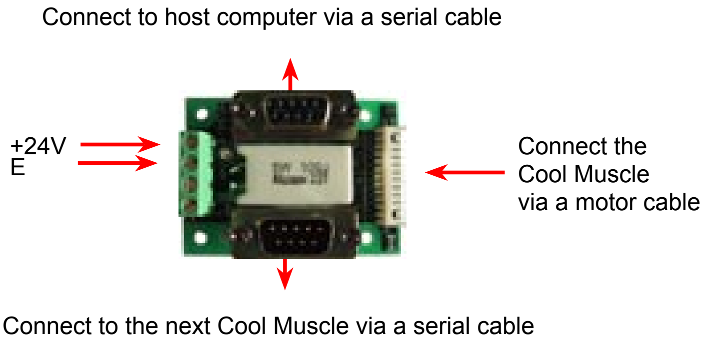
Network Card

ISerial nterface Card

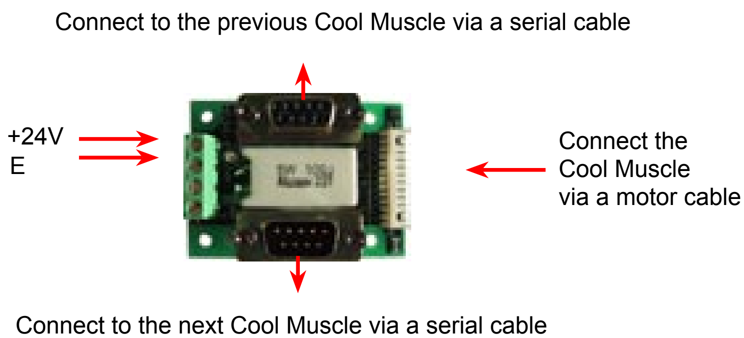


Network Card with Interface Card(Master Set)

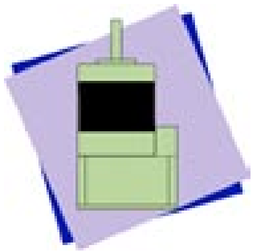
Connect the Network card as below.



Slave Motor



Note: Power may be transfered from card to card via the DB9 connectors as noted on page 16.



SECTION 4

INPUT AND OUTPUT

Pin Assignment and Layout

The Cool Muscle has 4 inputs and 2 outputs that are user definable. Each I/O port can be assigned multiple functions by setting motor parameters using CML. (Please refer to the Parameter section)





Connector Pin Layout

Pin#	Function	Return Pin	Assinable I/O types
1	DC+24V		N/A
2	GND-1		N/A
3	Input 2-		N/A
4	Output 2	11	Digital Out Analog Out Serial Transimit(TX)
5	Output 1	11	Digital Out Analog Out Serial Transimit(TX)
6	Input 4 (max 5V)	11	Digital In Analog In
7	Input 3 (max 5V)	11	Digital In
8	Input 1-		N/A
9	Input 2	3	Digital In Pulse Counter Serial Receive(RX)
10	input 1	8	Digital In Pulse Counter Serial Receive(RX)
11	GND2		N/A
12	DC+5 Out		N/A

Assignable functions - Input

Digital Input

Multiple functions can be assigned to a digital input port. A time delay can be set to create a ‘slow response’ signal, or a time delayed virtual signal based on a real signal (Quick Response signal) . Actions can be performed for the duration (or at the target voltage level) of the input or at both the rising and the falling edges of either the quick or the slow response signals. This provides 6 different states. Each state also has multiple features that can be assigned to it.

Assignable input functions at the target voltage level

#	Function
0	No Action
1	General Use
2	Origin Sensor
3	Manual Feed CW
4	Manual Feed CCW
5	Switch in-position signal to Index signal

Assignable input functions at the rising and <falling >edges

#	Function
0	No Action
1	Reset Alarm/Pause
2	Motor Free<Enable Motor>
3	Reset Counter
4	Execute next step
5	Execute previous step
6	Execute Bank 1
7	Go Origin
8	Manual Jog CW
9	Manual Jog CCW

Example;

Input 1 is configured to be

K28=0008

K30=0003

1. Manual Jog on the rising edge of a quick response signal.
2. Manual Feed for the duration of a slow response signal.

When a quick response input signal is detected, the motor will jog a distance defined by K50. When a slow response signal is detected, the motor feeds for the duration of the signal (K49).

Pulse Counter
CW/CCW
Step/Direction

Serial Input
Serial Communication

Analog In
Speed
Position

Input ports and assignable Input types

I/O	Digital In	Analog In	Pulse In	Serial In
Input 1	OK	NO	OK	OK
Input 2	OK	NO	OK	OK
Input 3(Max 5V)	OK	NO	NO	NO
Input 4(Max 5V)	OK	OK	NO	NO

OK: assignable
NO: not assignable

Assignable functions - Output

Digital Output

Assignable Output Functions

#	Function
0	Command
1	In-Position
2	Alarm
3	Analog Out
4	General Use 1
5	General Use 2
6	In-Position Output in Merge Mode

Analog Output

#	**Analog Output Types**
0	Target Position
1	Target Position magnified by 8
2	Current Position
3	Current Position magnified by 8
4	Position Error
5	Position Error magnified by 8
6	Current Velocity
7	Current Velocity magnified by 8
8	Current Torque
9	Current Torque magnified by 8

Serial Output

Serial Transmit to Host

Serial Transmit to Slave

Output ports and assignable Output types

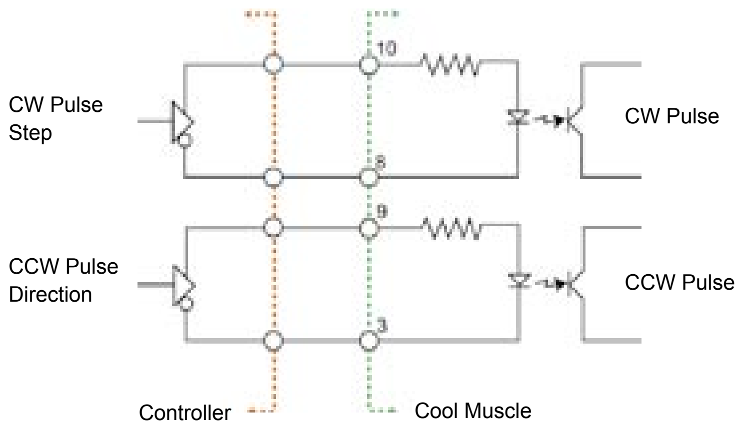
I/O	Digital Out	Analog Out	Serial Out
Output 1	OK	OK	OK
Output 2	OK	OK	OK

OK: assignable

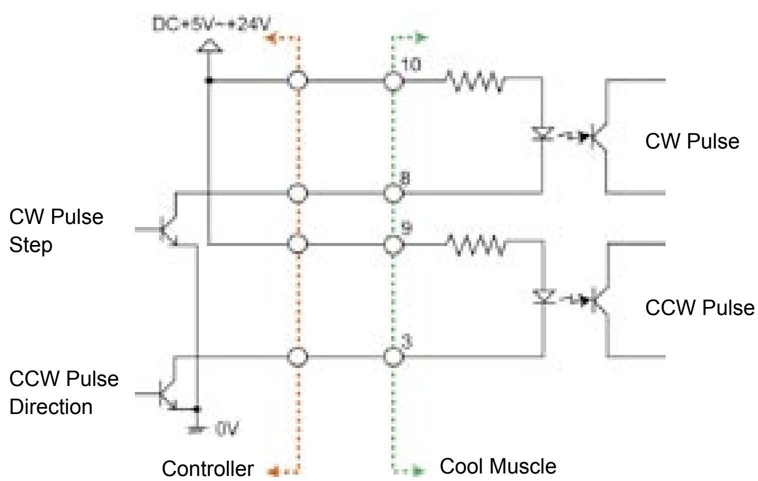
Wiring Examples

CW/CCW or Step/Direction

1. Linedriver



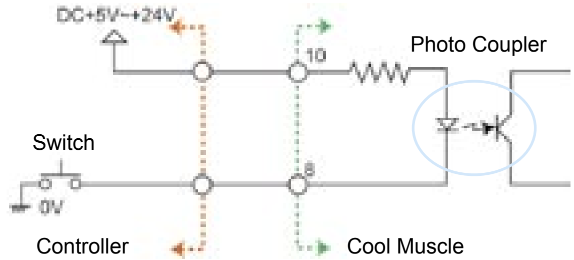
2. Open Collector



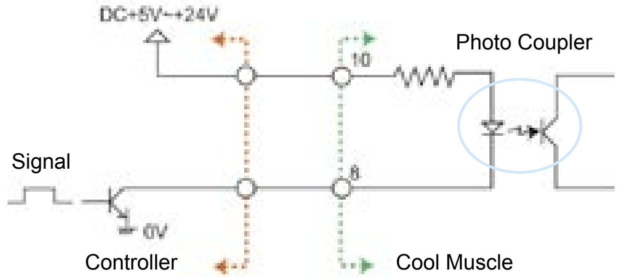
Digital Input :Switch and PLC

Input 1

Switch

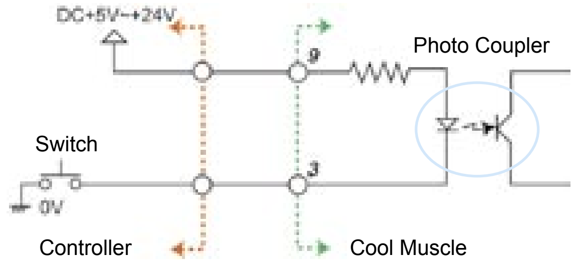


PLC

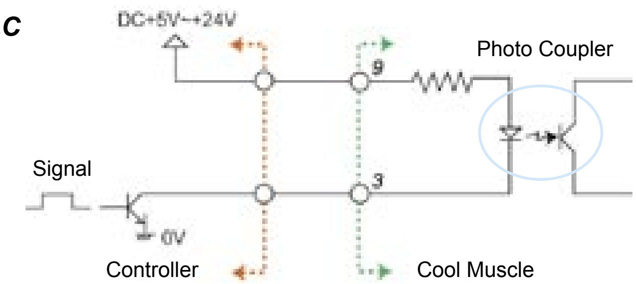


Input 2

Switch



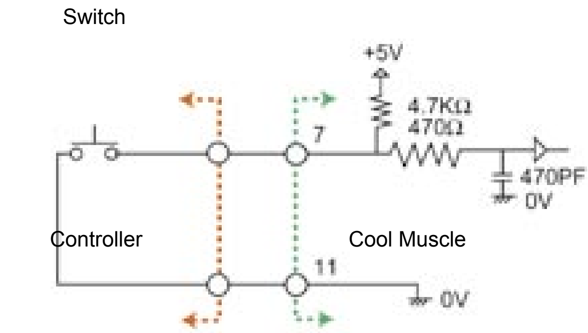
PLC



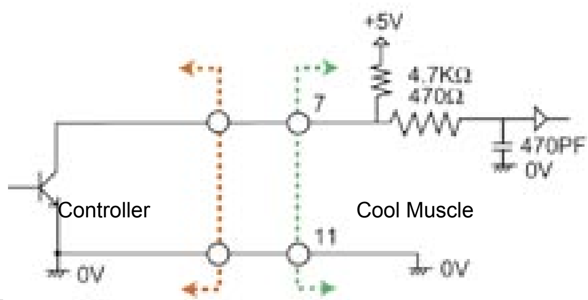
Digital Input : Switch and PLC
Analog Input

Input 3 (Max 5V)

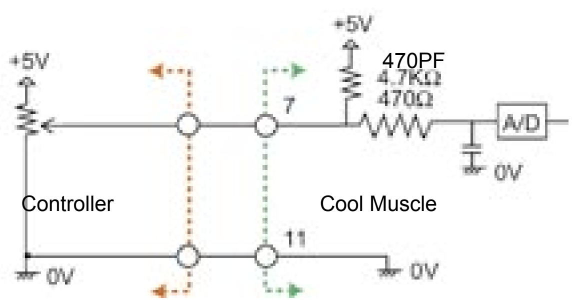
Switch



PLC

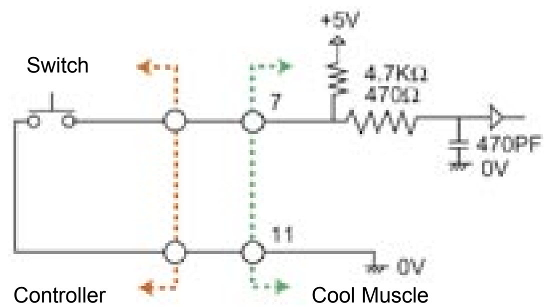


Analog

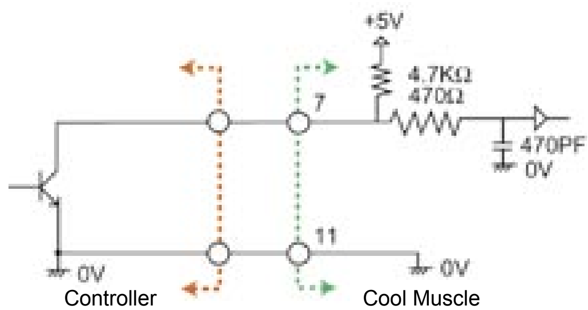


Input 4(Max 5V)

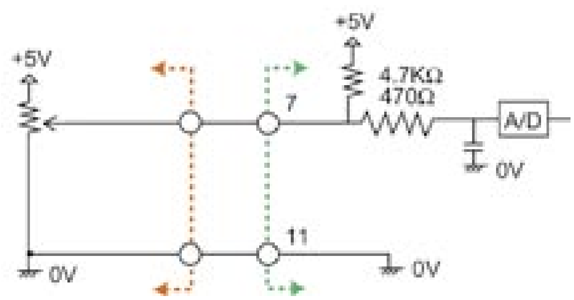
Switch



PLC



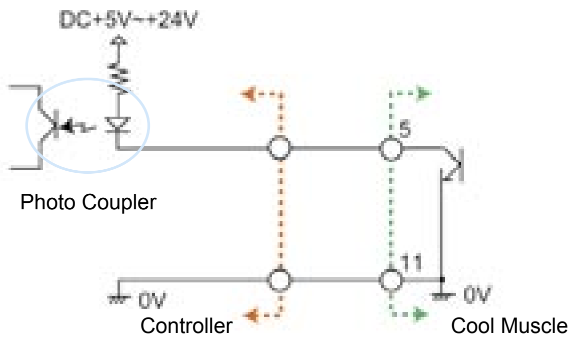
Analog



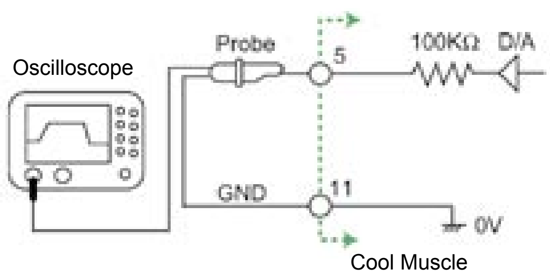
Output

Output1

Digital Output

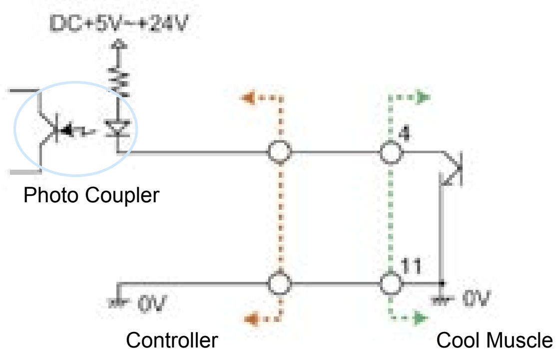


Analog Output

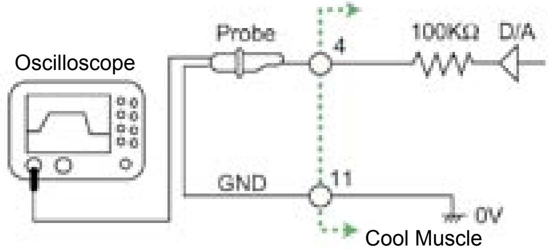


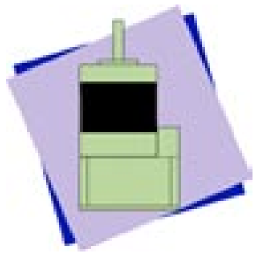
Output2

Digital Output



Analog Output



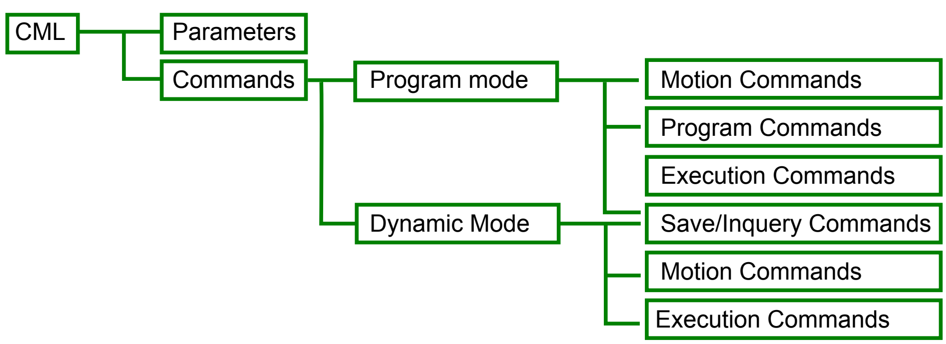


SECTION 5

CML STRUCTURE

What is CML?

CML(Cool Muscle Language) consists of a set of ASCII commands that provide a means to easily create motion programs for single or multi-axis systems. As the CML command structure allows commands to be identified by Motor ID, multiple networked Cool Muscles can be controlled via I/O or from a PC.



Parameters

Set parameters using CML. You can easily change parameters to suit your specifications.

Parameter Definition:
Parameter definitions take the form of:

K#=VALUE

K# signifies the parameter number to be set and VALUE represents the data to be assigned to K#. A list of parameters and their corresponding numbers are outlined in the Setting Parameter Section (Section 7).

ie.
K37=4
This parameters sets motor resolution to 2000 pulses per rotation.

Commands

There are two command modes:

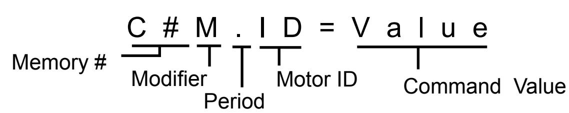
1. Program Command mode: used to create motion programs that can be downloaded to the Cool Muscle.

2. Dynamic Command mode: directly operate the motor via a PC to cause immediate action.

CML Structure

Command Structure and Rules

The basic structure of a CML command is:



Command Memory Location
Some commands have a limited number of memory locations. For example, Position and Speed can be stored in 25 and 15 memory locations respectively. Specify the memory location for the corresponding value.

P14.3=1000
The defined value 1000 is stored in position memory location 14 on Motor 3. The value set by the parameter is absolute.

Modifier
Modifiers exist in limited cases. The most common modifier is the relative motion modifier, + which is used with the P com-

mand. Adding + after the memory location makes the value relative. P14+.3 would command motor 3 to move relatively or incrementally the distance of 10000, which has been previously stored in Position Memory location 14.

Motor ID
When you wish to program multiple motors, you can specify a Motor ID as part of the CML command. The commands and parameters will be placed in the memory locations of the specified motor. When Motor ID is omitted, it is assumed to be either motor 1 or the last ID value used.

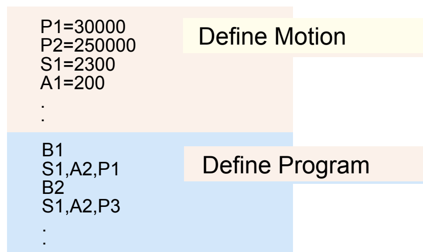
Value
The value defined in the command structure above is absolute. Values are typically used at the beginning of a CML program file to place values into the motors permanent memory. As an example, P14.3=10000 would place the value of 10000 into the Position memory location 14 of motor 3.

Spaces
Spaces should not be used in a command line.

Upper or Lower Case?
All the commands in CML program files are not case sensitive.

CML File Structure

CML files are made of 2 different parts. The first part defines memory contents while the second part defines motion programs. Some aspects of a motor's movement or control are defined by parameters. Speed and Acceleration data must be stated before the first target position. Subsequent moves will be based on previously stated Acceleration and Speed unless otherwise commanded.



Motion Definitions

Position, Speed, Acceleration and Timer definitions take the form of:

P#=VALUE
S#=VALUE
A#=VALUE
T#=VALUE

CML Program Banks

Program Banks are made up of CML Motion Commands. Program Banks can be stored in the motor's EEPROM. A program bank is defined as follows:

B# (carriage return)

B# specifies the program bank number. Each program bank can be regarded as an object. Program Banks may contain up to approximately 100 steps. CML Commands and their structures are outlined in the CML Command Listing.

CML Program File Example

K26=1111 / SET INPUT LOGIC
K49=15/SET MANUAL SPEED
K58=500/SET SOFTWARE LIMIT CW
K59=-500/SET SOFTWARE LIMIT CCW
P1=1000 / SET POSITION MEMORY CONTENTS
P2=2000
P3=5000
S1=200 / SET SPEED MEMORY CONTENTS
S2=1000
A1=200 / SET ACCELERATION MEMORY CONTENTS
A2=100
B1 /FIRST PROGRAM BANK
C2
C3 / JUMP TO THE BEGINNING OF PROGRAM BANK 3
B2 /SECOND PROGRAM BANK
S1,A2,P1 / SET SPEED, ACCELERATION AND POSITION
S2,P2
P3,P2,S1,P1 / PERFORMS A SMOOTH MOVEMENT FROM POINT TO POINT WITH STOPPING AND WITH A CHANGE IN SPEED BETWEEN P2 AND P1
B3 /THIRD PROGRAM BANK
S1,P3

(For more detailed information please refer to “Programming with CML”)

CML File Structure - Dynamic Command Mode

Dynamic Commands cause immediate action. This mode is typically used for debugging and non-repetitive movements.

Dynamic Command structure:

S=value
A=value
P=value
^ (execute)

In Program mode, you need to define position, speed, acceleration and timer. In Dynamic Command mode, you can directly drive the motor by entering motion profiles.

ie.
S=1000
A=200
P=10000
^

CML File/Command Downloading

There are a number of ways to use CML to set parameters, define motion, create programs, execute programs, drive motors and download programs onto the motor. Cool Works is free software designed to work with CML. You can also use Hyper terminal which usually is included with your Windows installation disk. Default Serial Communication specs are: 38.4Kbs, 1 Stop bit, No Parity, No Handshaking.

Changing Parameters, execute program banks, drive motor in dynamic command mode
Directly enter commands in a terminal program such as Cool Works or Hyper terminal.

Creating and editing program banks
It is recommended that a text editor is used to create and edit complicated program files. Make sure you save program files as a plain text file. You can download program files to the Cool Muscle via a terminal program by importing text files. You can also load parameters in the same method.

Any parameters specified in the file are changed in the motor without affecting unspecified parameters. All program banks are erased and replaced with the program banks specified in the CML file. When you want to write the program into the Motor's EEPROM, enter a "$". K Parameters are stored automatically.

SECTION 6

HOW TO USE COOL WORKS

Select COM port: This is where you select COM port that the Cool Muscle is connected to.

1. Start up Cool Works by clicking on the Cool Works icon.



2. The window below will appear. Select the COM port that the Cool Muscle using. By selecting the COM port, the Terminal Window/Setting Window/Jog Window buttons will be activated. In the Terminal window, you can communicate with tthe Cool Muscle directly. In the Motor Browser window, parameters specific to each motor type (P,C) can be easily set. The Jog Window lets you manually operate the Cool Muscle.



3. Check the RS-232 port to ensure the Cool Muscle is connected. Power the Cool Muscle.

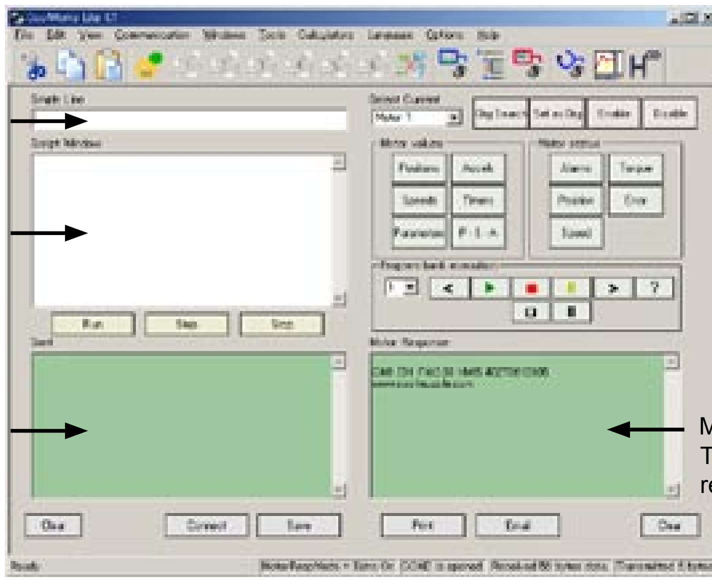
How To Use Terminal Window

After selecting the COM port, a teminal window will appear. In this window, you can manually set parameters, create motion programs, save to EEPROM, read parameters in the Cool Muscle, and operate the Cool Muscle in dynamic mode.

Single Line Window: Enter a single line command and press the Enter key.

Entry Window: You can type multiple line commands and edit text you enter. Once it is ready to be sent to the Motor, click “Run”. CML Script can also be executed from this window.

Sent Window:Commands and parameters you sent to the motor are displayed in this window.



Motor Response Window: This Window shows motor resonses.

Terminal Window Function Buttons

Go to Origin: Make the motor go origin (Positon 0)
Set as Origin: Set the current position to 0.
Enable motor: Enable motor in a motor free state.
Free Motor: Turn the motor into a motor free state.

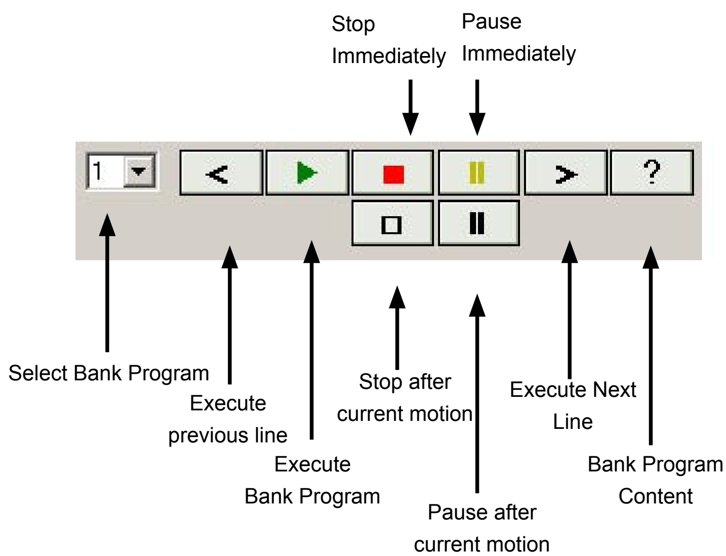
CML Values
Positions: displays posiiton data stored in the motor.
Speeds: displays speed data stored in the motor.
Accelerations: displays acceleration data stored in the motor.
Timers: displays timer data stored in the motor.
Dynamic Values: displays daynamic values of P, S, and A.
Motor settings: displays motor parameters from K25 to K63.

Current Status

Select Program Bank

Alarm: displays alarm state
Torque: displays current torque
Position: displays current position
Error: displays posiiton error
Speed: displays current speed

Execute Bank Program



Opening from existing CML files

You can open exiting CML files. Go the Menu Bar and select Files and Open. Select the file you want to use in Cool Works and click Open. The content of the file will be displayed in the Entry window. Click Send to send to the motor. Files need to be saved as a .txt file.

Switching to another window

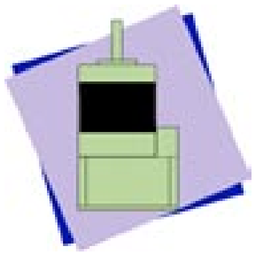
By clicking the icons on the menu bar, you can easily switch windows .

How Use Jog Window

In the Jog Window, you can manually operate the Cool Muscle. The max. speed, Acceleration are defined by the resolution and speed unit you set in this window.



Changing the resolution or speed unit in the Job Window immediatedly overwrites the existing setting in the motor. Hold and drag the speed scrol bar to operate the motor. To move the motor to position 0, click the Go Origin button.



SECTION 7

SETTING PARAMETERS

Before setting parameters

To set parameters on the Cool Muscle, you need to connect the motor to a PC via the Y cable or the network card. You can set parameters on multiple motors when the motors are network via network cables.

Using Y cable

Connect the Cool Muscle (refer to Section 3 “How to connect Cool Muscle”). Start up Cool Works or a terminal program and turn the motor on. Make sure that there is communication established between the Cool Muscle and your PC. (Please refer to Section 6 “How to use Cool Works”)

Using Network cable

Connect the Cool Muscle (refer to Section 3 “How to connect Cool Muscle”). Start up Cool Works or a terminal program and turn the motor on. Each motor is automatically given a motor ID. Make sure that there is communication established between the Cool Muscle and your PC. (Please refer to Section 6 “How to use Cool Works”)

Quick Reference Guide

A Quick Reference sheet is provided at the back of this manual, or as a PDF that can be downloaded from www.coolmuscle.com in the Download:Documents section

Parameter Structure

Parameters are defined at the beginning of a CML program file. Each parameter is assigned a memory location.

To set a parameter, follow the structure below.

K # =Value

Example;

K50=15
(Distance for Manual Jog is set to 15 pulses)

Parameters

This section provides detailed information on each parameter. For a quick reference please use the parameter summary table in Appendix 2.

Input/Output

Input Logic K26

Descriptions
This parameter sets the input logic.

Min.	***Max.***	***Default***	***Descriptions***
0000	1111	0000	0: normally open 1: normally closed

*Each number represents Input pin #4, 3, 2, 1 ordered from the right to left. ie. K26=0100
Input 4=normally open
Input 3=normally closed
Input 2=normally open
Input 1=normally open

The target voltage level can be either less than 0.8V or more than 3V. When input logic is set to true=low, the target voltage is less than 0.8, When it is set as true=high the target level is more than 3V(7mA min.).

Offset for Slow Response Signals K25

Descriptions

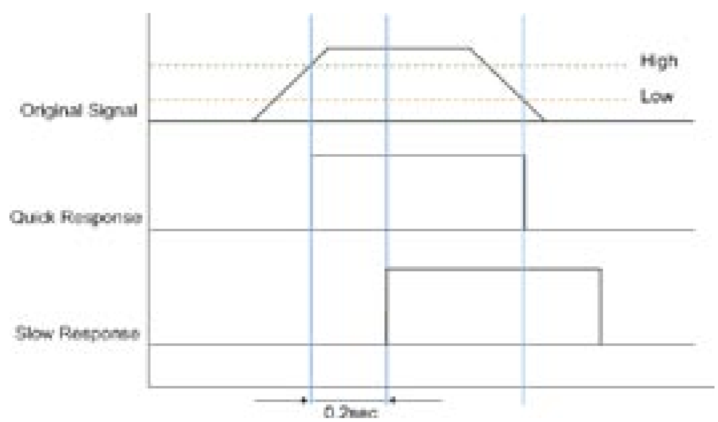
This parameter sets the time delay to create two sets of signals, quick response and slow response signals. If the signal is completed within the set time delay period, it is only recognized as a quick response signal and not recognized as a slow response signal.

Min.	*Max.*	*Default*	*Unit*
1111	9999	3333	0.1 sec

*Each number represents Input pin #4, 3, 2, 1 ordered from the right to left.
ie.K25=0100

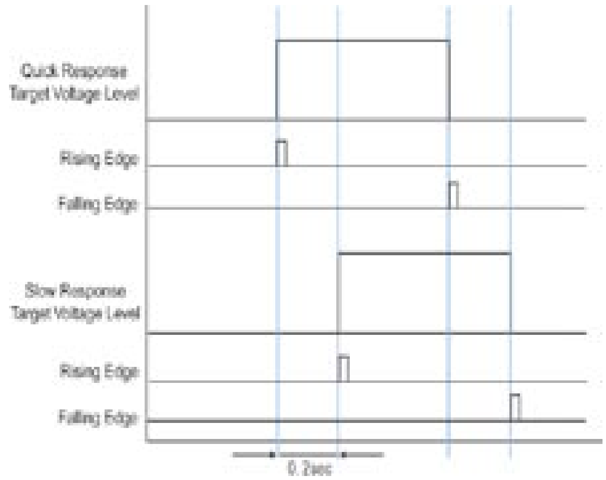
Quick and Slow response time offset

A quick response signal starts when it hits the high point. A slow response signal starts after the time offset set by the parameter (in this example it is set to 0.2sec).

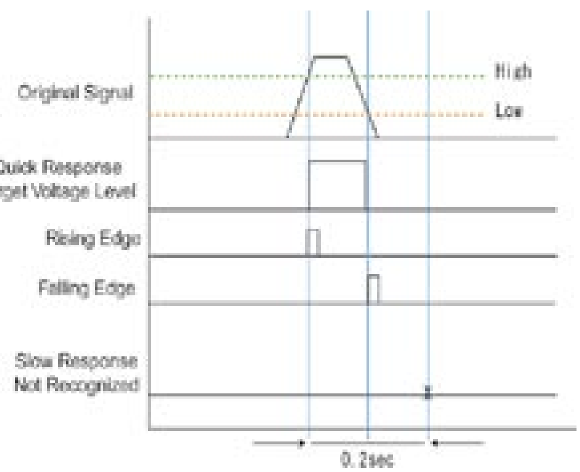


Quick and Slow response event triggers

Events (functions) can be assigned to the rising (and falling edges and the target voltage level of a quick and slow response signals.



Short Signal
When a signal is completed within the offset time only a quick response signal is recognized.



Input Functions at theTarget Voltage Level: Quick Response Signal (QTV)	K27

Input Functions at theTarget Voltage Level: Slow Response Signal (STV)	K30

Description
This parameter assigns a function performed for the duration of a signal. Assignable functions are listed in the table on the next page.

Assignable Functions for the target voltage level

#	Function	Descriptions
0	No Action	
1	General Use	Typically used for the I command. For example, B1 I2,C2,C3 When input 2 is true then execute bank2 otherwise execute bank3.
2	Origin Sensor	
3	Manual Feed CW	Motor turns in a CW direction for the duration of the signal
4	Manual Feed CCW	Motor turns in a CCW direction for the duration of the signal
5	Switch in-position signal to Index signal	When an output is set to in-position by parameter 34, it signals in-position when the motor reaches the target position. When the motor reaches the origin, it sends out index signal.
6	Execute Bank 1	Used to Execute Bank 1 on startup of motor
7	Emergency Stop	Stops the motor immediately and inhibits motion execution
8	Full Stop	

Min.	***Max.***	***Default***
0000	5555	0000(K27)
0000	5555	0000(K30)

*Each number represents Input pin #4, 3, 2, 1 ordered from the right to left.
ie. K27=0100

Input Function at the Rising Edge: Quick Response Signal(QR) — K28

Input Function at the Falling Edge: Quick Response Signal(QF) — K29

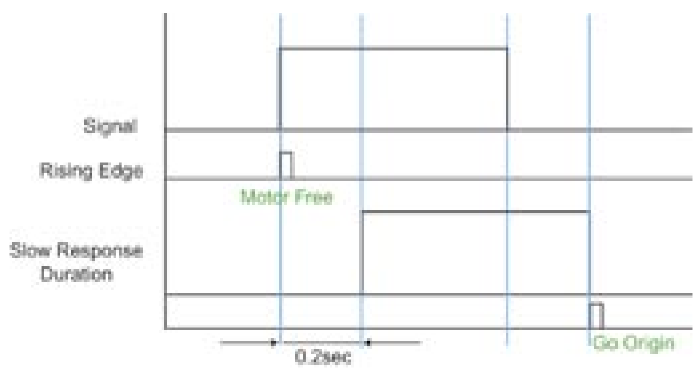
Input Function at the Rising Edge: Slow Response Signal(SR) — K31

Input Function at the Falling Edge: Slow Response Signal(SF) — K32

Functions that create motion should be limited to Input 3 and 4 for safety reasons. Inputs 1 and 2 are also used for serial communications. When connected to a serial port, the voltage used in the serial protocol might activate any function assigned to these ports.

Description

This parameter assigns a function performed at the edges of a signal. Functions or events should be assigned to event triggers in such a way as to make sense. For example, it may create undesirable movements if you assign Motor Free to the rising edge of a quick signal and Go origin to the duration of a slow response signal. With this input function assignment, when the motor is commanded to go back to origin, as it goes into a free state at the rising edge of a quick response signal, it will not be able to go back to origin.



Assignable Functions at the rising and falling edges of an input signal
Functions with < > indicate functions for the falling edge only

#	Function	Descriptions
0	No Action	
1	Reset Alarm/Pause	Reset Alarm:When alarm is on, it resets the alarm. Pause: Pauses the motor. Send a start signal to resume the motion.
2	Motor Free <Enable Motor>	Makes the motor go into a motor free state. <Enables the motor from a motor free state.>
3	Counter Reset	Makes the current position 0
4	Execute next step	Execute the next program line. This function is used when executing line by line within a program bank.
5	Execute previous step	Executes the previous program line.
6	Execute Bank 1	
7	Go Origin	
8	Manual Jog CW /Execute Bank 2	Motor turns in a CW direction the distance set by parameter 50. Set speed by parameter 49. Acceleration set by 42 (Also used for Origin Search acceleration) is applied. When K36 is set to 2 8 becomes Execute Bank 2 instead of Manual Jog CW
9	Manual Jog CCW /Execute Bank 3	Motor turns in a CCW direction the distance set by parameter 50. When K36 is set to 2 9 becomes Execute Bank 3 instead of Manual Jog CCW

Min.	***Max.***	***Default***
0000	9999	0100(K28)
0000	9999	0000(K29)
0000	9999	0200(K31)
0000	9999	0300(K32)

*Each number represents Input pin #4, 3, 2, 1 ordered from the right to left.

ie.K28=0100(Input 4=0, Input 3=1, Input 2=0, Input1=0)

Output logic K33

Descriptions
This parameter sets the output logic.

Min.	***Max.***	***Default***	***Description***
00	11	11	0: Normally Open 1: Normally Closed

*Each number separated by a comma represents Output pin #2,1 ordered from the right to left. When typing in parameters, you don’t need commas. ie. K33=01

Output Functions K34

Description
This parameter assigns a function to an output. Assignable functions are as below.

#	**Function**	**Descriptions**
0	Command	
1	In-Position	Sends out an in-position signal when the motor reaches the target position.
2	Alarm	
3	Analog Out	Outputs analog waves for monitoring. Choose a type of information by parameter 35.
4	General Use1	
5	General Use 2	
7	Timed Trigger	Outputs a signal at certain intervals. Timing is set by K24
8	Motor Free	
9	Torque Limit Reached	For Push mode

Min.	***Max.***	***Default***
00	66	21

*Each number separated by a comma represents Output pin #2,1 ordered from the right to left.
ie.K33=01

Interval for Timed Trigger K24

Description
This parameter lets you set the distance interval for timed trigger .

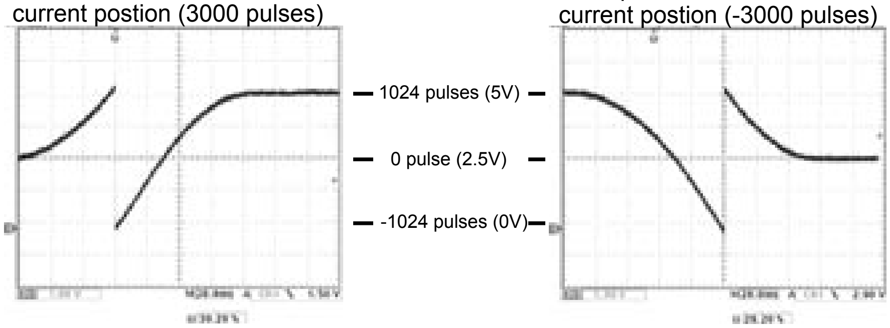
Min.	*Max.*	*Default*	*Unit*
10	32767	1000	pulses

Analog Outout Functiona K35

Description
This parameter lets you select an analog output type that you can monitor with an oscilloscope. Make sure that you have selected Analog Out by parameter 34. The range of analog output is 0-5 V. 2.5V is the baseline. The overall value is cummuitive but the analog output is displayed in the set range (0-5V).

Example :
current postion (3000 pulses)

Example :
current postion (-3000 pulses)



#	Analog Output Types	Unit
0	Target Position	±1024 pulses/±2.5V
1	Target Position manified by 8	±128 pulses/±2.5V
2	Current Position	±1024 pulses/±2.5V
3	Current Position manified by 8	±128 pulses/±2.5V
4	Position Error	±1024 pulses/±2.5V
5	Position Error manified by 8	±128 pulses/±2.5V
6	Current Velocity	±2400rpm pulses/±2.5V
7	Current Velocity manified by 8	±300rpm pulses/±2.5V
8	Current Torque	±9.3kgfcm pulses/±2.5V(CM1-x-23L20) ±5.1kgfcm pulses/±2.5V(CM1-x-17L30)
9	Current Torque manified by 8	±1.16kgfcm pulses/±2.5V(CM1-x-23L20) ±0.64kgfcm pulses/±2.5V(CM1-x-17L30)

Min.	*Max.*	*Default*
00	99	31

*Each number separated by a comma represents Output pin #2,1 ordered from the right to left. ie.K34=21

Resolution

Motor Resolution K37

Parameter values between 20 and 30 should be used when extreamly slow speeds are required.

NEW!
K37=100

Improvements in V2 firmware now allow for a resolution of 50000 with a 1pps speed unit.

Description
This parameter sets the motor's resolution (pulses/rotation). This parameter also sets the speed unit used by the speed command (S). 0-10 sets the speed unit to 100pps and 20-30 sets the speed unit to 10pps.

Speed Unit 100pps				**Speed Unit 10pps**			
#	Resolution	#	Resolution	#	Resolution	#	Resolution
0	200	6	5000	20	200	26	5000
1	400	7	10000	21	400	27	10000
2	500	8	25000	22	500	28	25000
3	1000(default)	9		23	1000	29	
4	2000	10	50000	24	2000	30	50000
5	2500			24	2500		

Setting K37=100 sets the resolution to 50000 and the speed unit to 1pps. The entire range of speed is acheivable with this setting.

Homing/Origin Search

The Cool Msucle will stop during origin search if it does not reach an origin within 5 minuite or 80 rotations.

There are two types of origin search routines.

1. Mechanical origin: determined either by an origin switch or bumper.
2. Electrical origin: set by the offset distance from the mechanical origin. Typically used as a home position or starting point.

Return to Origin Speed K42

Description
This parameter sets the speed when the motor goes back to origin.

Min.	***Max.***	***Default***	***Unit***
1	5000	10	100pps

Origin Search/Manual Feed Acceleration K43

Description

This parameter sets the acceleration used when the motor returns to origin. This acceleration is also used for manual feed.

Min.	***Max.***	***Default***	***Unit***
1	5000	100	$Kpps^2$

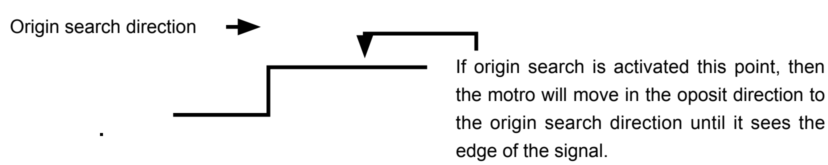
Origin Direction K45

Description

This parameter sets the direction that the origin is located.

#	Direction
0(Default)	CW
1	CCW

If origin search is activated at the point where the origin switch is on, the motor will move in the oposit direction until it sees the edge of the origin switch signal



Origin Search Method K46

Description

This parameter specifies the method used during an origin search. Origin can be determined using hard-stop/bumper or an origin switch. The motor hits the bumper at very low speed and keeps pushing until it reaches a specified current level (preset by parameter 47). When the current level has reached the set current level, the motor determines that it has found origin. This method eliminates the need for origin switches.

You need to set the following related parameters.

Stopper:
Go Origin speed(K42)
Acceleration(K43)
Current Level(K47)

Origin Switch:
Go Origin speed(K42)
Acceleration(K43)

#	Method
0(Default)	Stopper/Bumper
1	Stopper/Bumper(Automatically start origin search when powered on.)
2	Origin Switch
3	Origin Switch(Automatically start origin search when powered on.)

Current level when searching for origin using a stopper — K47

Description
Sets the current level at which the motor will determine that the origin position has been reached during a hard-stop origin search routine. It is set by a percentage of the peak current of the motor.

Min.	***Max.***	***Default***	***Unit***
10	100	30	% of the peak current

Offset Distance Between Machine Origin and Electrical Origin — K48

Description
This parameter sets the offset distance between the mechanical and electrical origin. This parameter is used when you want the motor to find the origin and automatically go to the electrical origin (starting point). If you set the parameter to 0 (default), the

motor stays at the mechanical origin. The speed for moving from the mechanical origin to the electrical origin is set to 2 times as fast as the speed for return to origin.

Min.	Max.	Default	Unit
-32767	32767	0	*100 pulses

In-position

In-position signal K55

In-position Signal is sent out when the motor is within the error tolerance set by K55. If the motor is running at a very low speed and the error tolerance is large, you may see in-position signals even before the motor reaches the target position.

Description

This parameter sets the error tolerance for an In-Position signal. When the motor is within the specified error tolerance, then an in-position signal will be sent out.

Min.	Max.	Default	Unit
1	100	5	pulses

Alarms and Limits

Error Counter Over Flow K56

Description

This parameter sets a max. value for the error counter. If the error exceeds the set value, then the motor goes in to a motor free state.

Min.	Max.	Default	Unit
1	32767	50	Kpulses

Overload time K57

Description
This parameter sets the time delay between the detection of overload and alarm signal output. If you set an even number, the motor will ignore regenarated voltage alarm.

Min.	*Max.*	*Default*	*Unit*
100	10000	3000	msec

Software Overrun protection + side K58

Software Overrun protection – side K59

Description
This parameter sets the overrun protection on both CW and CCW sides. When the motor hits the overrun point it immediately stops. When the distance is set to 0 (default) the overrun setting is ignored.

Min.	*Max.*	*Default*	*Unit*
0	327670	0	*100 pulses(K58)
-32767	0	0	*100 pulses(K59)

Deceleration

Deceleration ratio K44

Description
This parameter sets the deceleration based on acceleration (defined by the “A” command). The deceleration can be set as a percentage of acceleration. Note that the percentage set by this parameter will be applied to all deceleration in the CML program files. If you wish to set deceleration independent from acceleration you need to use CML commands. (please refer to the command section)

Min.	*Max.*	*Default*	*Unit*
10	500	100	%

Manual Motion

Manual Jog speed K49

Description
This parameter sets the rotational speed for manual jog/feed.

Min.	*Max.*	*Default*	*Unit*
1	5000	10	100 pps

Manual Jog distance K50

Description
This parameter sets the travel distance for manual jog.

Min.	*Max.*	*Default*	*Unit*
1	100	10	100 pulses

Gain Adjustment

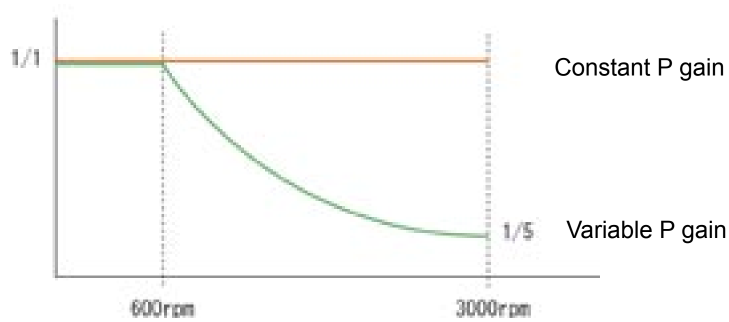
Position P Gain K52

With V2 firmware, the Cool Muscle now uses a modern control theory called H Infinity.

Please refer to Appendix 9 to learn about tuning the new system.

K52-K54 are no longer used in V2 or greater firmware

Description
This parameter sets the position loop proportional gain.There are two modes; constant and variable Position P gain. Enter an even value to apply constant P gain. Enter an odd value to automatically vary P gain depending on speed. Variable gain is typically useful when using a belt driven system.

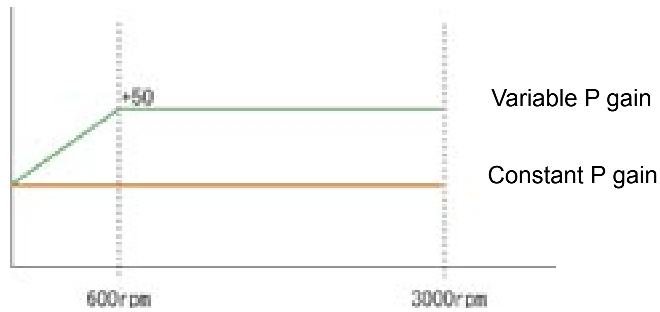


Variable P gain starts dropping at 600RPM. At 1200RPM, 1/5 the value set will be applied.

Min.	Max.	Default	Unit
0	512	depens on the motor type	1/256

Speed P Gain K53

Description
This parameter sets the speed loop proportional gain.There are two modes; constant and variable Speed P gain. Enter an even value to apply constant P gain. Enter an odd value to automatically vary P gain depending on speed.



Variable P tain increases gradually until the motor reaches 600RPM. The set value is increased by 50 at 600RPM.

Min.	Max.	Default	Unit
0	512	depens on the motor type	1/256

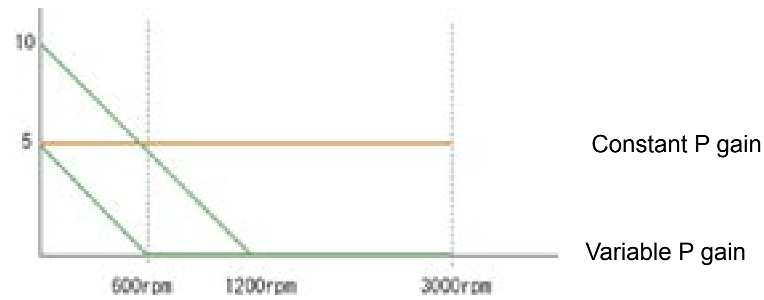
Speed I Gain K54

Description
This parameter sets the speed loop proportional gain.There are two modes; constant and variable Speed I gain. Enter an even value to apply constant P gain. Enter an odd value to automatically vary I gain depending on speed.

In the variable I gain mode,the relationship between the speed at which I gain will not be applid (Value 0) and the set I gain are as below.

Speed (I gain becomes 0) = set I gain x 120

When the I gain is set to 5, the speed that I gain becomes 0 is 600RPM. If the I gain is se to 10, the speed that the I gain becomes 0 is 1200RPM.



Min.	*Max.*	*Default*	*Unit*
0	512	depens on the motor type	1/1024

Pulse Type Cool Muscle

Pulse type K36

Description
This parameter sets Pulse Type Cool Muscle to either CW/CWW or Pulse/Direction. If you are uing C Type Cool Muslce, this parameter allows you to assign Exeute Bank 2 and 3 to rising and falling edges of input signals.

P type Cool Muscle

#	*Type*
0(default)	CW/CCW
1	Step/Direction

C type Cool Muscle

#	*Type*
2	changes the assignlable input functions, Jog Manual CW and Jog Manual CCW, to Execute Bank 2 and 3. (K28,29,31,32)

When the motor is set to semi-closed loop, the Cool

Analog Input functions are now part of the C Type motor interface.

The V type Cool Muscle has been discontinued.

Analog Input Control

Analog Interface Type K38

Description

This parameter sets the motor to either Speed or position control when K64 is set to 9.

V type Cool Muscle

#	Type
0(default)	Speed
1	Position

Analog Speed Range K40

This parameter sets the speed at 4.8V for Speed type analog Cool Muscle. Increase an analog input voltage from 2.6 to 4.8V to increase the speed in a CW direction. Decrease an analog input voltage from 2.4V to 0.2V to increase the speed in a CCW direction. Note that an area between 2.4V and 2.6V is a dead zone within which no motion will occur.

Min.	Max.	Default	Unit
200	4000	3000	rpm

Analog Distance Range K41

An increase in an analog input voltage from 0V to 4.8V changes position in proportion to the input voltage. This parameter sets the maximum travel distance at 4.8V.

Example: Resolution 2000 (set by parameter 37)
Maximum movement range 90 degrees
2000/(90/360 degrees)=500 pulses

Example : Resolution 2000 (set by parameter 37)
Pitch 5mm
Maximum travel distance 30mm

30mm/5mm=6rotations
6rotations x 2000= 12000 pulses

Min.	*Max.*	*Default*	*Unit*
-32767	32767	2000	pulses

Low Pass Filter

Voltage Filter Gain K39

Description
Cut-off frequency for Low Pass Filter for AD converter.

Min.	*Max.*	*Default*	*Unit*
0	1024	128	5(rad/sec)

(unit: 5000[times/sec]/1024=5[rad/sec])

Push Mode

Push Current K60

Description
Current level applied when the motor is in push mode. The current level is set by % of peak current.

Min.	*Max.*	*Default*	*Unit*
10	70	50	%

Push Time K61

Description
The time that the motor keeps pushing for when it is in push mode.

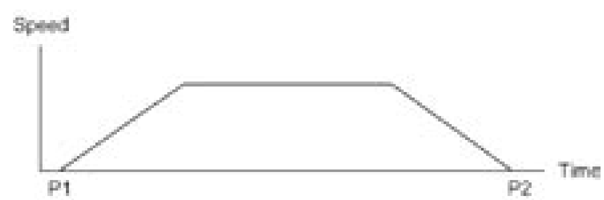
Min.	*Max.*	*Default*	*Unit*
10	3000	200	msec

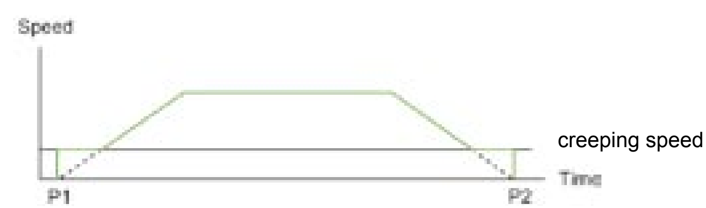
Creeping Speed

Creeping Speed K51

Description
Sets creeping speed. You can adjust motor response time by changing creeping speed. Setting creeping speed too high may cause the motor to vibrate.

Min.	*Max.*	*Default*	*Unit*
1	1000	10	100pps





Instead of gradually reaching the target speed or comes to a stop, the motor quickly reaches the target speed or stops, improving motor response time.

Communication Speed

Baud Rate K20

Description
This parameter sets the baud rate that your PC and the Cool Muscle communicate at.

#	Rate
0(default)	38.4k
1	9.6k
2	19.2k
3	57.6k

Semi-Closed/Full-Close Loop Mode

Semi-Close Loop Vector Angle K21

Description
You can set the Cool Muscle to either semi-closed or full-closed loop. In semi-closed loop mode, the Cool Muscle becomes open loop during holding, eliminating hunting problems. This parameter allows you to switch between full colose loop mode. When the Cool Muscle is set to Semi-Closed Loop, you can use this parameter to set the vector angle at which the Cool Muscle returns to closed loop.

#	Rate
0(default)	full Closed Loop
1~36	# x 0.1 degree

Tiriger Timing for semi-closed loop mode K22

Description

This parameter lets you set a triger timing for semi-closed loop. This trigger timing is a time delay between in-position signal and the moment that the Cool Muslce becomes open loop.

Min.	***Max.***	***Default***	***Unit***
10	1000	100	msec

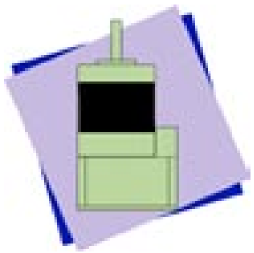
Motor Status Report Style

Polling/Event Driven Mode K23

Description

This parameter lets you select a motor status report style; Polling or Event Driven.

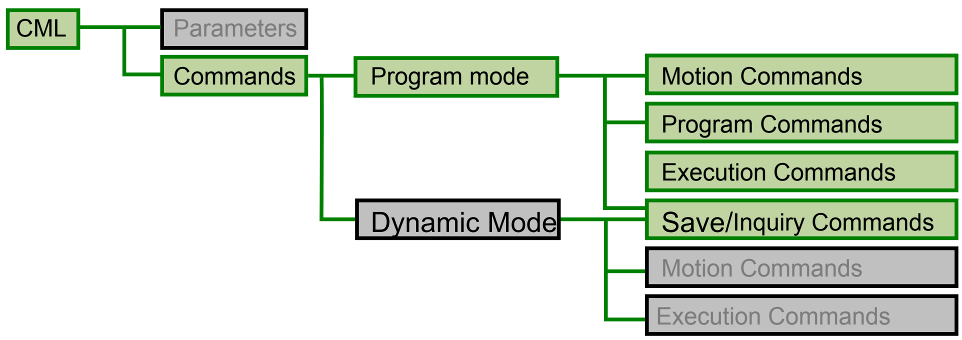
#	***style***
0	Polling
1	running status, Overflow, Over Load, In Position, Circle function, Pushing Torque
2	Input Status (represented as a HEX value)
4	Output Status
8	Status for Start, Stop, Enable, Disable, Pause



SECTION 8

PROGRAM MODE COMMANDS

This section contains CML commands for creating program banks along with examples. Commands for creating and executing programs banks, query commands are explained in Section 8.



Motion Commands - define motion

Positions, Speeds, Accelerations and Timers need to be defined before motion is executed.

Position Definition

Description P

This parameter can define multiple positions. (Max. 25) The positions set by the P command is absolute. To make the value relative, add + after memory location. Once positions are defined, you can call the stored position. If the value is larger than 1000000000 the motor will turn continuously CW as defined by speed setting. To turn the motor continuously in a CCW direction wet the speed to a negative value.

Examples:

P23=2500 (sets P23 to 2500)
P=1000000000 (sets the motor to turn continuously)
P23+ (commands the motor to go relatively/incrementally the distance of 2500)

Memory#	***Min.***	***Max.***	***Unit***
25	0	999999999	pulses

Speed Definition S

Description

This parameter can define multiple speeds. (Max. 15) The speed unit can be set to either 100pps or 10pps by parameter K37.

Examples:

S10=20(sets S10 to 2000pps)

Memory#	***Min.***	***Max.***	***Unit***
15	1	32767	100 pps or 10pps

Acceleration Definition A

Description

This parameter can define multiple accelerations. (Max. 8) The accelerations set by this command are absolute. This commands in combination with K44 also defines deceleration.

Examples:

A2=200(sets A2 to 200Kpps^2)

Memory#	***Min.***	***Max.***	***Unit***
8	1	32767	Kpps^2

Timer Definition T

Description

This parameter can define multiple timers. (Max. 8)

Examples:

T1=1000(sets T1 to 1000msec)

Memory#	***Min.***	***Max.***	***Unit***
8	1	32767	msec

Program Commands - create programs

After parameters and motion values are defined, you can start creating motion programs. The command structure using the P, S, A, and T is as below.

S1,A1,P1 (go to P1 with the speed and acceleration defined by S1 and A1)

Motion commands are separated by a comma. The speed, and acceleration that you want to apply to reach the target position must be placed before the P command.

The maximum steps that each motor can store is 100. Each command is counted as one step. For example, the above example is consisted of 3 steps.

Program Bank | B

Description
This command defines the beginning of a program bank. It also defines a program bank number. The maximum number of program banks that you can use is 16. Note that when you enter the B command, all the existing programs will be erased.

Example:
B1 (beginning of program bank 1)
S1,A1,P1 (Go to position 1 at the speed and acceleration defined by S1 and A1)
S2,A1,P3 (Go to position 3 at the speed and acceleration defined by S2 and A1)

Call Program Bank

C

When the program bank called by the C command is completed, it goes to the next line of the line in which the C command was executed.

Description

This command calls and executes specific program banks. The following program will execute Bank 1 and once Bank2 is completed it will execute bank 3, and then come back to the first line of Bank 1, which will execute Bank 2.

Example:

B1 (beginning of program bank 1)
C2 (call B2 and execute)
C3(call B3 and execute)
B2 (beginning of program bank 2)
S3,A1,P1
B3 (beginning of program bank 3)
S2,A2,P3

Jump to Program Bank

J

Description

This command jumps to and executes specific program banks. The following program will execute Bank 1 and jump to and execute Bank2 . The program ends at the end of Bank 2. It will not go back to the program bank in which the jump command was originally executed.

B1 (beginning of program bank 1)
A1,S1,P3
J2 (jump to B2 and execute)
B2 (beginning of program bank 2)
x10(loop 10 times)
S3,A1,P1
P2
(It does not go back to B1)

Input | I

Description
This command makes the motor perform specific actions or program banks based on the specified input status. This command takes the following form;

I#,True,False

Example:
B1 (beginning of program bank 1)
I1,C3,C2 (If Input 1 is true execute bank 3, otherwise execute bank 2)

Using the I and T0 commands you can create nesting withing banks. Nesting can be used to select banks. The maximum nesting level is 10. A nesting example is described in the “applied CML program examples” section.

No Action | T0

Description
This command is used with the I command.

Example:
B1
I1,C2,T0(zero)
If Input 1 is true, the motor executes bank 2, otherwise it does nothing. When the I command line is followed by another command line, T0 moves the cursor to the next line and executes it. When bank 1 is completed, it goes back to line 1 in Bank 1. When Input 1 is true, it jumps to Bank 2 and execute it. Once B2 is completed, it goes back to line 2 in Bank 1.

Input 1 is False

B1
I1,C2,T0 (line 1)
A1,S1,P3 (line 2)
B2
A1,S1,P3

Input 1 is Ture

Set Output High — O

Set Output Low — F

Description
The O command makes the specified output high/on and the F command turns the specified output low/off.

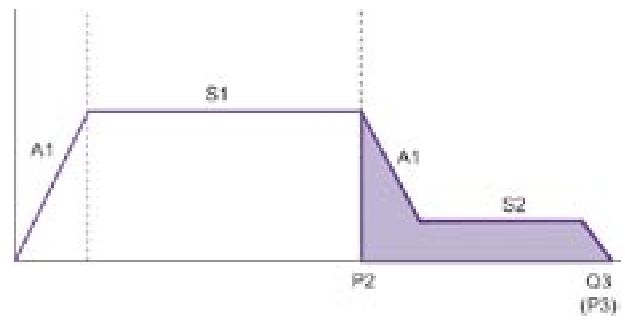
Example:
B1 (beginning of program bank 1)
S1,A2,P1,O1,P2,F1,P3
(this program commands the motor to go to P3 without stopping at P1 and P2. Output 1 goes high when the motor passes P1 and goes low when it passes P2.)

Push Mode — Q

Description
The Q command takes the position value defined in a position memory location and commands the motor to perform a push mode.
A push mode is a special type of motion designed for applications typically performed by pnematic cylinders. When moving to a final destination if the motor encounters resistance it may trigger an alarm. In push mode, a certain amount of resistance at the final destination is expected or anticipated and the motor does not produce an alarm. The motor based on the parameter settings will push into or apply pressure at the destination for a given time at a set current level or torque. You can make the motor push an object continuously by setting parameter K61 to 3001. Stop the motor from pushing by executing a new action using commands such as [, >, or |.

Example:
B1 (beginning of program bank 1)
S1,A1,P2,S2,Q3
(The motor performs a push mode between P2 and P3 at the

speed defined by S2 and for the time and at the current level defined by push mode parameters)



Shaded area represents a push mode

Loop X

Description
You can specify the number of times that you want a specific program bank to loop. A value of O=infinite loops. Not specifying an X command results in a single execution of the program.

Example:
B1,X3 (Loop B1 three times)
A1,S1,P1
P2

P command without a wait Y

Description
When the Cool Muscles are networked, a master motor waits for in-position signals from slave motors before it executes the next step in a program bank (with the P command). Using the Y command instead of the P command, you can make networked motors to execute motion without waiting for other motors.

Example:
B1
A1.1,S1.1,P1.1,A1.2,S1.2,A1.2,P1.2
P2.1,Y2.2 (Motor 2 execute the next line without waiting for in-position signal from Motor 1)

Q command without a wait | Z

Description
When the Cool Muscles are networked, a master motor waits for in-position signals from slave motors before it executes the next step in a program bank (with the P command). Using the Z command instead of the Q command, you can make networked motors to perform push mode action without waiting for other motors.

Example:
B1
A1.1,S1.1,P1.1,A1.2,S1.2,A1.2,P1.2
P2.1,Q2.2 (Motor 2 performs push mode action without waiting for in-position signal from Motor 1)

Comment | /

Description
You can enter comments in the CML program files. Comments can be entered after a “/”. Comments entered by “/” are not downloaded to the Cool Muscle. They are used for your reference in a text file.

Execution Commands

Execute Program Bank | [

Description
Execute the specified program Bank.

[1 (executes Bank 1)

Execute Next Line >

Description

Execute line by line in a program bank. This command is useful when debugging programs. Entering > will execute the first line in Bank 1. Enter > again to execute the next line. To execute a line in banks other than bank 1, you need to execute the bank program using the [command and stop. Then enter >.

Execute Previous Line <

Description

Execute previous line.

Go origin |

(it is a bar not an alphabet “I”)

Description

Goes back to or searches for origin. Origin method is set by parameter 46.

| makes the motor do origin search.

|1 makes the motor go to position 0.

|2 sets the current position to 0.

Pause]

Description

] pauses the motor. Enter [bank# to resume the program. Note that the program is resumed with the next executable line.

B1
S1,A1,P1
P2
P3

If the] command is entered when the motor is moving from P1 to P2, the motor will stop somewhere inbetween the two points.

If [1 is entered to resume the program, third command line will be executed, making the motor direcly go to P3.

](CR)] pauses the motor and resets the program to its first command line.

Stop after Completing Motion }

Description
Stops after completing current motion.

Motor Free)

Description
De-enegizes the motor windings, leaving the rotor free to rotate. The motor`s circuit board and encoder remain active.

Enable motor (

Description
Use this command to enable the motor in a free state.

Save to EEPROM $

Description
Saves CML Program Banks and P, S, A, and T values to the EEPROM on the Cool Muscle. Changes to K parameter values are automatically saved when changed.

```
B1
S1,A1,P1

P1=1000
S1=100
A1=100
$
```

Query ?

Description

Shows motor information.

?#	Content	
?0-16	bank no.	
?70	input status (D0~D7)	Displays Inputs = high 1=input 1 2=input2 4= input 3 8=input 4 Multiple inputs: add numbers ie. input 1 and 3 are high=5 1 + 4 =5 If the total is more than 10 A=input 2 and 4 B=input 1,2 and 4 C=input 3 and 4 D=input 1,3 and 4 E=input 2,3,and 4 F=input 1,2,3 and 4
?71	input status (D8~D15) reserved for expansion	
?72	output status (D0~D7)	Displays Outputs = high 1=Output 1 2=Output 2
?73	analog input value CH1	
?74	analog input value CH2	
?75	counter value CH1	
?76	counter value CH2	
?90	parameter list for user	
?91	position definition list	
?92	speed definition list	
?93	acceleration definition list	
?94	timer definition list	
?95	position error	
?96	current position	
?97	current speed	
?99	Motor status message	0: Running 1: overflow 2: overspeed/regenerative voltage 4: overload 8: Inposition

Basic CML Program Examples

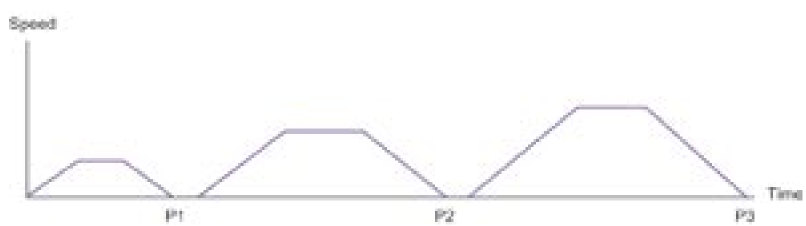
Most machine movement is repetitive and makes use of only a few positions, speeds, accelerations and timers. This type of information makes up part of the header of a CML file.
The following motion definitions are used for the examples in this section.

P1=20000
P2=72000
P3=120000
P4=500000
S1=5500
S2=7000
S3=8500
S4=10000
A1=200
A2=500
A3=800

Basic Point to Point motion

B1
A1,S1,P1
S2,P2
S3,P3

This creates the motion below.

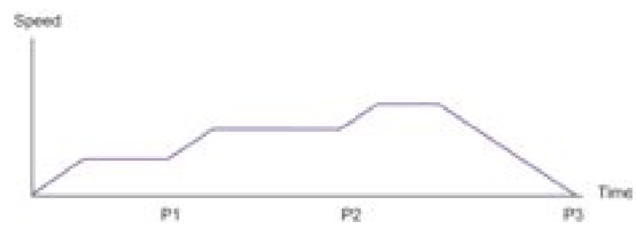


Starting at the origin position, the motor moves from point to

point, stopping and starting at each point with equal acceleration reaching different speeds.
Notice that the same acceleration is applied to P1, P2 and P3.
You can leave out S and A when the same Speed or Acceleration are applied to different target positions, saving the number of steps.

Continuous Point to Point motion with different speeds

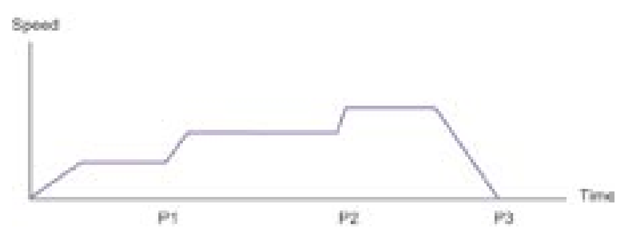
B2
A1,S1,P1,S2,P2,S3,P3
By combining the same motion commands on one line, the following motion is created.



As compared to program bank B1, B2 has no stops in its motion between Origin and P3. Speed was changed at each point and the acceleration and deceleration remained equal throughout.

Continuous Point to Point motion with different speeds and accelerations

B3
A1,S1,P1,A2,S2,P2,A3,S4,P3

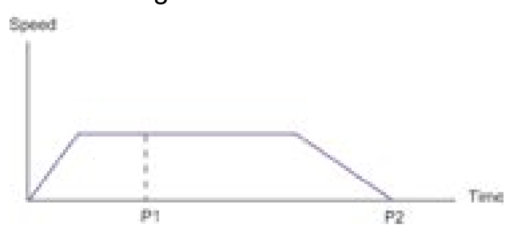


Including acceleration changes at each point and increasing the speed prior to the last point may create motion similar to the chart below.

Point to Point motion with different deceleration

The deceleration rate is always equal to the last acceleration rate specified taking into account the value of parameter K44. This can be modified in two ways. The first way is to set the Acceleration/Deceleration Ratio parameter. This results in is a set ratio for every motion. The second way is to place multiple commands within a single line of CML, it is possible to specify both an acceleration and a deceleration. Suppose the final destination is P2 and a quick acceleration followed by a slow deceleration is required.

B5
A1,S1,P1,A2,P2
Resulting in the following motion.

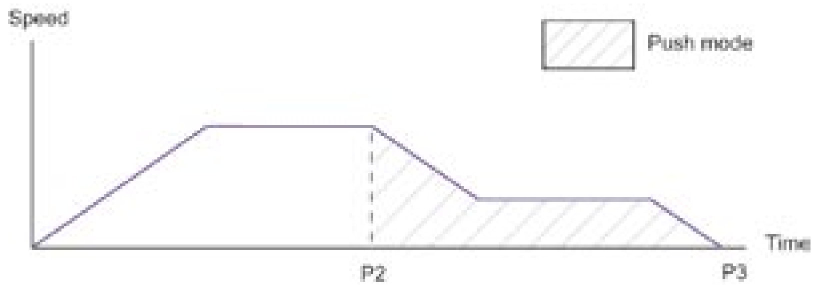


Keeping the speed across two points equal, and changing the acceleration value will give you a different deceleration to reach the final destination.

Push Mode

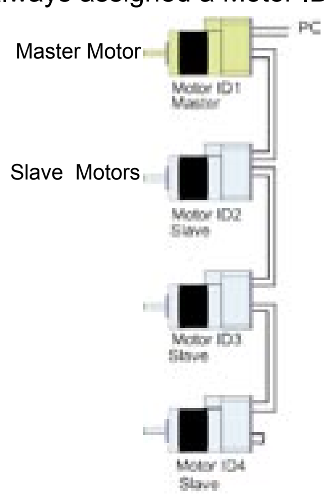
Using the Q command makes the motor perform a push mode. A push mode mimics a typical pneumatic cylinder and would keep pushing for a given time and at a set current level when the motor finds a resistance such as stopper or bumper. Parameters K60 and K61 also need to be set to values best fitting your requirments.

B6
A1,S3,P2,S1,Q3



Networking Cool Muscles

Multiple Cool Muscle motors can be connected in a serial network that communicates via the RS-232 or RS-485 protocol. In this configuration, CML commands are passed from motor to motor along the device network. At power up, each Cool Muscle is assigned a unique Motor ID which is numbered consecutively from the first (Master) motor on the network. Note that the Master motor is always assigned a Motor ID of 1.



In the standard RS-232 network configuration, only the Master motor is capable of executing stored Program Banks. Consequently, all motion routines which are to be perfomed by Slave motors must be stored in the Master motor. **Note:** Motion parameters (P, S, A) called by the Master motor in a Program Bank must be defined in the memory of the referenced Slave motor.

Defining Motion for Multiple Motors

As previously mentioned, a unique Motor ID is assigned to each Cool Muscle motor on a device network. Each Motor ID is used to reference a particular motor when motion commands are executed or parameters are defined for that motor. Commands and parameters are directed to an individual motor by attaching the corresponding Motor ID as a postfix to any CML statement in the fashion outlined below.

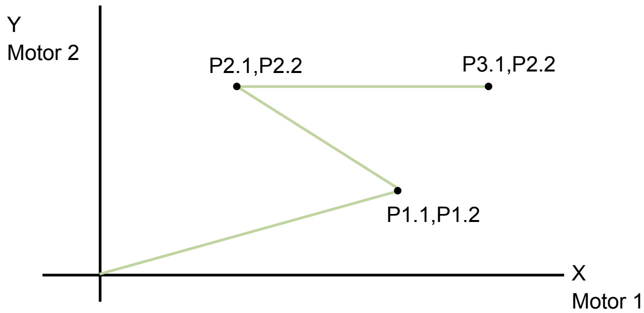
P1.1=1000 (Set Position 1 on Motor 1 to 1000 pulses)
P1.2=2000 (Set Position 1 on Motor 2 to 2000 pulses)
K45.1 = 0 (Set Motor 1 Origin Direction to Clockwise)
K45.2 = 1 (Set Motor 2 Origin Direction to Counter-Clockwise)
S1.1=200 (Set Speed 1 on Motor 1 to 2000 pulse/sec)
S1.2=400 (Set Speed 1 on Motor 2 to 4000 pulse/sec)
$.1 (Save Motor 1 parameters to EEPROM)
^.2 (Motor 2 Dynamic Command Execution)

When using Cool Works to define multiple motion parameters for a particular networked motor, there is a convenient alternative to the Motor ID postfix style of addressing. To direct a consecutive series of CML definitions to a Cool Muscle with Motor ID **N**, simply precede the block of statements with the command "**.N**" Until another Motor ID is asserted in this manner, all CML commands will be directed to the motor with ID **N**. For clarity, an example is given below.

.1
P1=1000 (Set Position 1 on Motor 1 to 1000 pulses)
K33 = 00 (Set Motor 1 Output Logic to Normally Open)
$ (Save Motor 1 parameters to EEPROM)
.2
P1=2000 (Set Position 1 on Motor 2 to 2000 pulses)
K33 = 11 (Set Output Logic on Motor 2 to Normally Closed)
$ (Save Motor 2 parameters to EEPROM)

Creating Network Motion Programs

In order to create a motion program for networked Cool Muscles, the Master motor must contain Program Banks which define motion profiles for all Slave motors. An example of this in practice is laid out below.



B1.1
A1.1, S1.1, P1.1, A1.2, S1.2, P1.2
P2.1,P2.2
P3.1

Line 1

Begin definition of Program Bank 1 on Motor 1 (Master).

Line 2

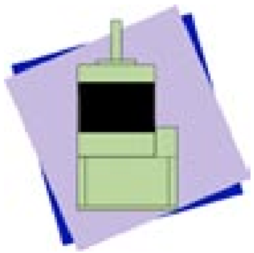
In network Program Banks, commands directed to different motors in the same line of CML code are executed simultaneously. In this case, Motor 1 and Motor 2 will start to rotate towards their respective target positions (P1.1, P1.2) at the same time. Each motor will accelerate to its individual desired speed (S1.1, S1.2) at the defined rate (A1.1, A1.2).

Line 3

Once each motor has reached its initial target position, the motion commands defined in Line 3 are executed simultaneously. Both motors move to their respective positions (P2.1, P2.2) at their individual speed and acceleration defined in Line 2.

Line 4

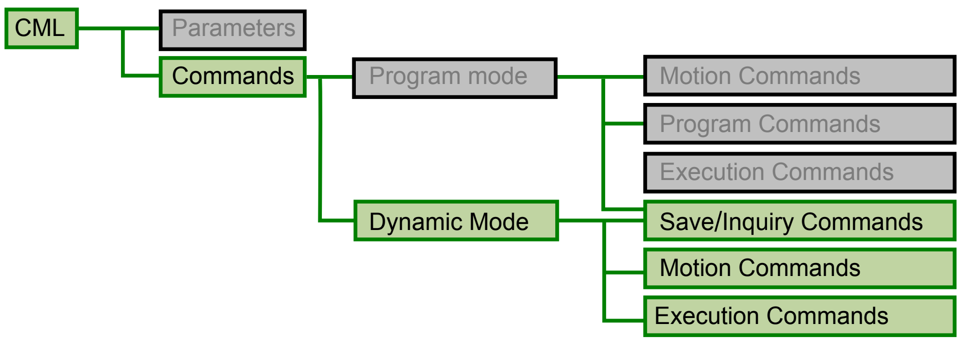
After both motors reach their target position (P2.1, P2.2) Motor 1 rotates to position P3.1 and Program Bank 1 ends.



SECTION 9

Dynamic Mode Commands

In this section, motion commands, execution commands for dynamic mode are introduced.



Motion Commands

Dynamic Commands cause immediate action. This mode is typically used for debugging and non-repetitive movements.

Position — P

Description
Set position.

P=10000

Unit: Pulses

Speed — S

Description
Set Speed.

S=250

Unit: 100pps or 10pps set by parameter K37.

Acceleration	A

Description
Set Acceleration.

A=250

Unit: Kpps^2

Execution Commands

PTP movement	^

Description
Directly type in commands to cause immediate PTP movement.

P=1000000

S=2000

A=200

^

CP movement	~

Description
This command makes the motor to move between specified points at the maximum speed with maximum acceleration. The values are incremental.

P=2

~

P=5

~

P=10

~

Go Origin |

(it is a bar not an alphabet “I”)

Description
Goes back to or searches for origin. Origin method is set by parameter 46.

| makes the motor do origin search
|1 makes the motor go to position 0.
|2 sets the current position to 0.

Pause]

Description
] pauses the motor. Enter [bank# to resume the program. Note that the program is resumed with the next executable line.

B1
S1,A1,P1
P2
P3

If the] command is entered when the motor is moving from P1 to P2, the motor will stop somewhere inbetween the two points. If [1 is entered to resume the program, third command line will be executed, making the motor direcly go to P3.

](CR)] pauses the motor and resets the program to its first command line.

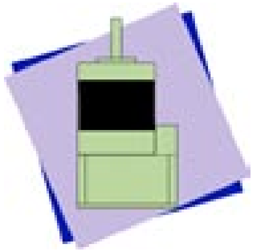
Motor Free)

Description
De-energizes the motor windings, leaving the rotor free.

Enable motor (

Description
Energizes the motor windings and resets any error states after an alarm.

When executing this command after an alarm, please first send a] to stop any program execution. Enabling the motor after an alarm or after a) command issued during operation may result in immidate resumption of motion.

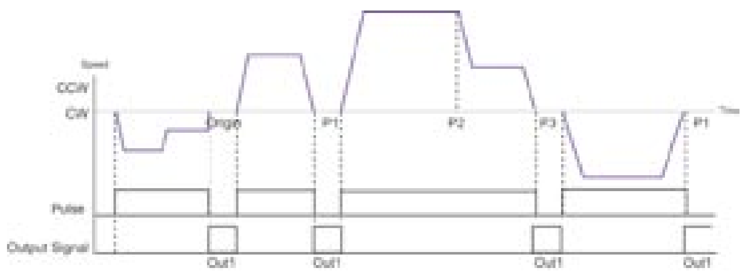


SECTION 10

Applied CML Program Examples

This section shows you applied CML program examples using the parameters and commands.

Example 1: Setting I/Os-Pulse type



Parameters
K25=1122(set time delay between quick and slow response to 0.1sec, 0.1 sec, 0.2sec, 0.2sec for input 4,3,2,1)
K26=1111 (Set input logic for port 4,3,2,1 to high in true)
K27=0052 (Set QTV funcitons)
K31=0210(Set SR functions)
K32=0200(Set SF functions)
K33=00(set output logic for port 2,1 to high in true)
K34=21(set output functions)

Input

	QTV	SR	SF
Input 1	Origin Signal		
Input 2	Index Signal	Alarm Reset	
Input 3		Motor Free	Enable Motor

Output

Output 1	Inposition
Output 2	Alarm

Motion descriptions

This example shows how you can assign multiple functions to a single input. K25 sets the time delay between quick and slow response signals. Origin is searched or defined by an origin switch. When the target voltage level on Input 2 is high, it switches output 2 from In-position signal to index signal. This is how you can differentiate in-position from on-origin signals.

Example 2: Search origin using a bumper - Pulse type



Parameters

K25=1122(set time delay between quick and slow response to 0.1sec, 0.1 sec, 0.2sec, 0.2sec for input 4,3,2,1)

K26=1111 (Set input logic for port 4,3,2,1 to high in true)

K27=0050 (Set QTVfuncitons)

K28=0007 (Set QR funcitons)

K31=0210(Set SR functions)

K32=0200(Set SF functions)

K33=11(set output logic for port 2,1 to high in true)

K34=21(set output functions)

K46=0 (find origin by stopper/bumper)

K42=15 (set Origin search speed)

K43=150(set origin Search acceleration)

K45=0 (directin of origin set to CW)

K47=30 (current level when pushing a bumper)

Input

	QR	QTV	SR	SF
Input 1	Origin Search	Origin Signal		
Input 2		Index Signal	Alarm Reset	
Input 3			Motor Free	Enable Motor

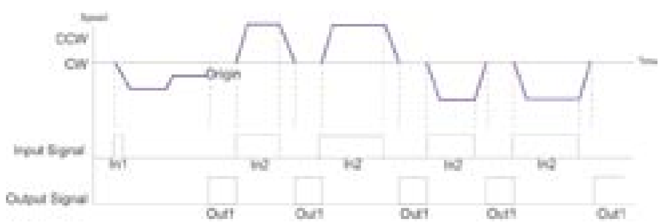
Output

Output 1	Inposition
Output 2	Alarm

Motion descriptions

The previous example showed how to search for origin using an origin sensor. As an alternative, you can use a bumper or stopper to make the motor search for an origin. The motor starts to search for an origin when input 1 goes high. (at the rising edge of a quick response signal). The motor turns in a CW direction at the speed and with the acceleration set by parameter 42 and 43. The motor determines that it has found or reached an origin when its current reaches 30% of its peak current.

Example 3: Manual Feed and jog



Parameters

K25=1122(set time delay between quick and slow response to 0.1sec, 0.1 sec, 0.2sec, 0.2sec for input 4,3,2,1)

K26=1111 (Set input logic for port 4,3,2,1 to high in true)

K28=0198(Set quick rising funcitons)K30=0043(set STV functions)

K30=0043(Set STV functions)

K33=00(set output logic for port 2,1 to high in true)

K34=21(set output functions)

K46=1 (find origin by switch)

K42=15 (set speed at which the motor searches or goes to origin)

K43=150(set acceleration with which the motor searches for or goes to origin)

K45=0 (direction of origin set to CW)

K49=15 (set Manual speed)
K50=20 (set Manual jog pulses)

Input

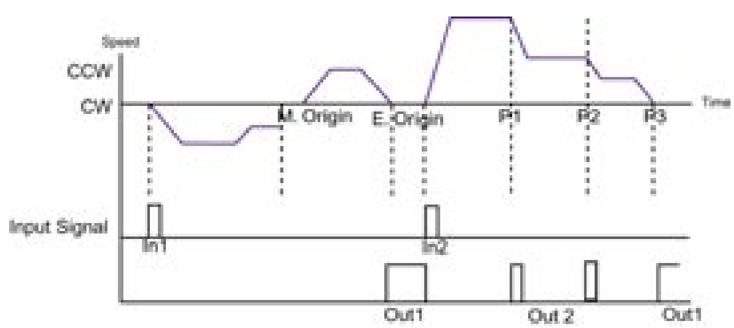
	QR	STV
Input 1	Manual Jog CW	Manual Feed CW
Input 2	Manual Jog CCW	Manual Feed CCW
Input 3	Alarm Reset	

Output

Output 1	Inposition
Output 2	Alarm

Motion descriptions
This examples shows manual jog and manual feed. The travel distance for the manual jog is set to 20 pulses by parameter 50. The manual speed is set to 1500 pps by parameter 49. The acceleration for manual operation is the same as the acceleration of go origin. A short signal (less than 0.2sec) will trigger manual jog which turns the motor 20 pulses. If a signal longer than 0.2 is sent, the motor turns continuously for the duration of the signal. In this case, manual jog and manual feed are merged.

Example 4: Electric orientation with continuous PTP program



Parameters
K25=1122(set time delay between quick and slow response to 0.1sec, 0.1 sec, 0.2sec, 0.2sec for input 4,3,2,1)
K26=1111 (Set input logic for port 4,3,2,1 to high in true)

K27=0002(Set STV functions)
K28=0167(Set QR functions)
K33=00(set output logic for port 2,1 to high in true)
K34=21(set output functions)
K46=1 (find origin by switch)
K42=15 (set speed at which the motor searches or goes to origin)
K43=150(set acceleration with which the motor searches for or goes to origin)
K45=0 (direction of origin set to CW)
K48=100 (set mechanical and electric origin offset to 10000 pulses)

Input

	QR	STV
Input 1	Origin Search	Origin Signal
Input 2	Execute 1	
Input 3	Pause/Alarm reset	

Output

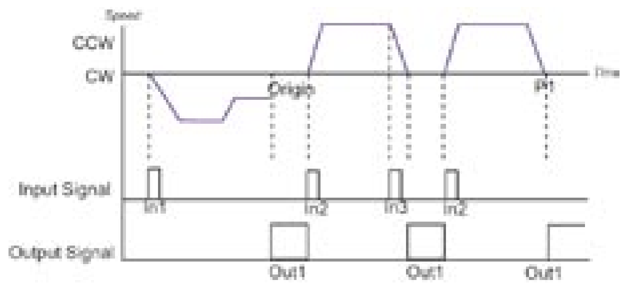
Output 1	Inposition
Output 2	General Use

Motion program
B1
S1,A1,P1,S2,P2,S3,P3

Motion descriptions
By setting the offset distance between the mechanical and electric origins, you can make the motor automatically move to the electric origin after it finds or goes to a mechanical origin. In this example, the electric origin is set to 10000 pulses away from the mechanical origin.
A program in Bank 1 is executed at the target voltage level of a quick response signal on Input 1. Output 2 goes high when the motor passes P1 and stays high until it passes P2. This type

of signal is different from in-position signals and can be used to signal your controller or other motors to perform tasks.

Example 5: Pause



Parameters

K25=1122(set time delay between quick and slow response to 0.1sec, 0.1 sec, 0.2sec, 0.2sec for input 4,3,2,1)

K26=1111 (Set input logic for port 4,3,2,1 to high in true)

K27=0002(Set QTV level functions)

K28=0761(Set QR functions)

K33=00(set output logic for port 2,1 to high in true)

K34=21(set output functions)

K46=1 (find origin by switch)

K42=15 (set speed at which the motor searches or goes to origin)

K43=150(set acceleration with which the motor searches for or goes to origin)

K45=0 (direction of origin set to CW)

Input

	QR	QTV
Input 1	Origin Search	Origin Signal
Input 2	Execute Bank1	
Input 3	Alarm Reset/pause	

Output

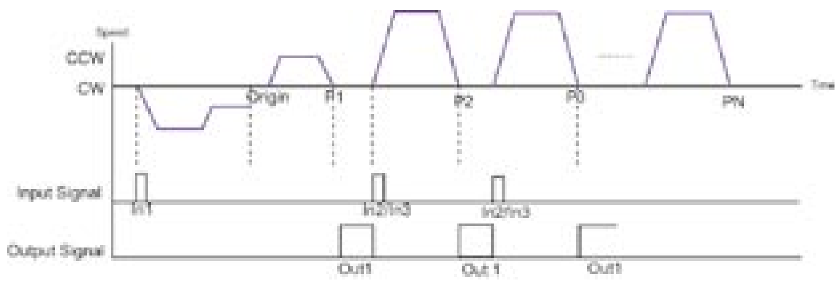
Output 1	Inposition
Output 2	Alarm

Motion program
B1

S1,A1,P1

Motion descriptions
This example shows how you can pause a program. During the execution of Bank 1, the motor is paused somewhere inbetween the origin and P1. By making Input 2 high, the paused program will be resumed.

Example 6: Bank Select using nesting



Parameters
K25=1122(set time delay between quick and slow response to 0.1sec, 0.1 sec, 0.2sec, 0.2sec for input 4,3,2,1)
K26=1111 (Set input logic for port 4,3,2,1 to high in true)
K27=0112(Set QTV functions)
K28=2667(Set QR functions)
K29=2000(Set QF level functions)
K33=00(set output logic for port 2,1 to high in true)
K34=21(set output functions)
K46=1 (find origin by switch)
K42=15 (set speed at which the motor searches or goes to origin)
K43=150(set acceleration with which the motor searches for or goes to origin)
K45=0 (direction of origin set to CW)

Input

	QR	STV	QF
Input 1	Origin Search	Origin Signal	
Input 2	Execute Bank 1	general use	
Input 3	Execute Bank 1	geenral use	
Input 4	Motor Free		Enable Motor

Output

Output 1	Inposition
Output 2	Alarm

Motion program

B1
T1
I2,C2,C3
B2
I3,C4,T0
S2,A1,P3
B3
I3,C5,T0
S1,A3,P2
B4
S3,A1,P4
B5
S1,A1,P1

Motion descriptions

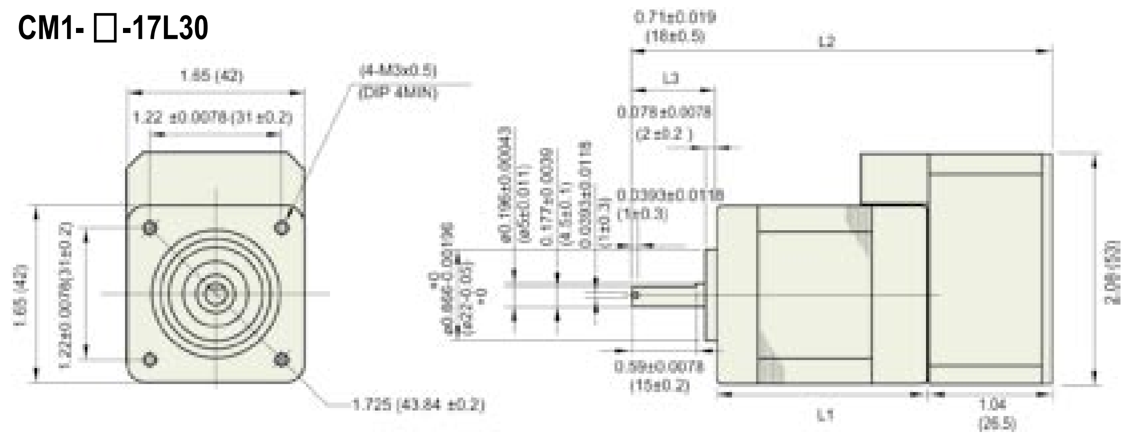
You can select a bank via digital Inputs by using a nesting technique. The above motion program shows how you can program in a such a way as to allow for bank select using two input ports. When both Input 2 and Input 3 are low Bank 1 will be executed. The motor will go to P1 at the speed and acceleration defined by S1 and A1. When Input 2 is high and input 3 is low, it jumps to Bank 2 and excutes the program. When Input 2 is low and Input 3 is high, it jumps to Bank 3 and executes it. When both inputs are high, then it jumps to Bank 4 and executes the program.

	Input 2	Input 3
Bank 1	Low	Low
Bank 2	High	Low
Bank 3	Low	High
Bank 4	High	HIgh

Appendix1

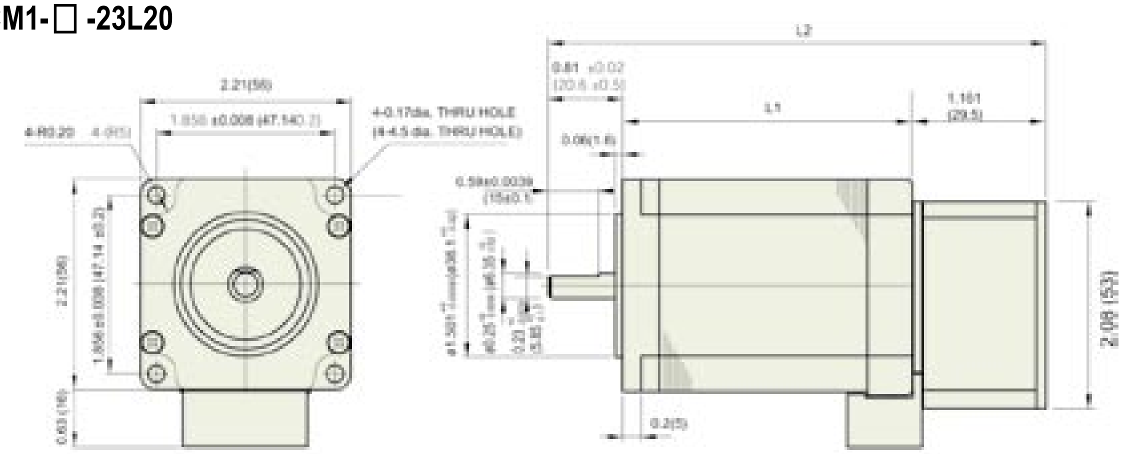
Cool Muscle Dimensions

CM1- □-17L30



Model	L1	L2	L3
CM1-x-17L30	1.968(50)	3.74±0.039 (95±1)	0.708±0.0196 (18±0.5)
CM1-x-17S30	1.338(34)	3.326±0.039 (34.5±1)	0.944±0.0196 (24±0.5)

CM1-□ -23L20



Model	L1	L2
CM1-x-23L20	2.992(76)	4.964±0.039 (126.1±1)
CM1-x-23S30	1.653(42)	3.625±0.039 (92.1±1)

Cool Muscle Specifications

CM1- □-17□30

Specification	CM1- X -17L30	CM1- X -17S30
Motor Output Power	18W	18W
Max. Speed	3000rpm	3000rpm
Rated Continuous Torque Nm(oz.in)	0.38(53.8)	0.082(11.61)
Rated Peak Torque Kgfcm Nm(oz.in)	0.54(76.4)	0.117(16.56)
Load Inertia Allowance $Kg.cm.s^2$ ($ozf.in.s^2$)	$7.6x10^{-4}$($1.08x10^{-2}$)	$3.8x10^{-4}$($5.39x10^{-3}$)
	Depending on the load inertia,servo gain needs to be adjusted. Within the above range: adjustable by parameters Outside the range: please contact us.	
Motor Inertia $Kg.cm.s^2$ ($ozf.in.s^2$)	$7.6x10^{-5}$ ($1.08x10^{-3}$)	$3.8x10^{-5}$($5.39x10^{-4}$)
Encoder	Incremental Magnetic Encoder (50000 pulses/rotation)	
Control Method	Closed Loop Vector Control	
Input Supply Voltage	DC24V±10%	
Input Supply Current Rated (Continous/Rated peak)	1.2A/1.8A	1.2A/1.8A
Resolution Pulse Rotation (pulse/rotation)	200,400,500,1000,2000,2500,5000,10000,25000,50000 Set by parameter	
Environmental Conditions Operation/Storage	between 0 ºC and 40ºC/between -20ºC and +60ºC	
Operating Humidity	Less than 90%RH	
Impact/Vibration	Less than 10G/Less than 1G	

CM1- □-23□30

Specification	CM1- X -23L20	CM1- X -23S30
Motor Output Power	30W	45W
Max. Speed	2000rpm	3000rpm
Rated Continuous TorqueKgfcm(Nm)	0.87(123.2)	0.29(41.06)
Rated Peak Torque Kgfcm(Nm)	1.24(175.6)	0.46(65.14)
Load Inertia Allowance $Kg.cm.s^2$ ($ozf.in.s^2$)	$4.61x10^{-3}$($6.5x10^{-2}$)	$1.4x10^{-3}$($1.9x10^{-2}$)
	Depending on the load inertia, servo gain needs to be adjusted. Within the above range: adjustable by parameters Outside the range: please contact us.	
Motor Inertia□$Kg.cm.s^2$ ($ozf.in.s^2$)	$4.61x10^{-4}$($6.5x10^{-3}$)	$1.4x10^{-4}$($1.9x10^{-3}$)
Encoder	Incremental Magnetic Encoder (50000 pulses/rotation)	
Control Method	Closed Loop Vector Control	
Input Supply Voltage	DC24V±10%	
Input Supply Current Rated (Continous/Rated peak)	2.6A/3.4A	3.9A/5.1A
Resolution Pulse Rotation (pulse/rotation)	200,400,500,1000,2000,2500,5000,10000,25000,50000 Set by parameter	
Environmental Conditions Operation/Storage	between 0 ºC and 40ºC/between -20ºC and +60ºC	
Operating Humidity	Less than 90%RH	
Impact/Vibration	Less than 10G/Less than 1G	

Pin Layout

			Default			
PIN #	Type	Function	CW/CCW	Step/Dir	Analog	Computer
1	+24VDC IN	Power				
2	GROUND 1					
3	INPUT 2-		CCW -	Direction -		
4	OUTPUT 2+	Didigal Out Analog Out Serial TX				Serial Port2
5	OUTPUT 1+	Didigal Out Analog Out Serial TX				Serial Port1
6	INPUT 4+	Digital Input, Analog Input				
7	INPUT 3+	Digital Input			V +	
8	INPUT 1-		CW-	Step -		
9	INPUT 2+	Digital Input,Pulse Counter,Serial RX	CCW +	Direction +		Serial Port2
10	INPUT 1+	Digital Input,Pulse Counter,Serial RX	CW +	Step +		Serial Port1
11	GROUND 2				V -	
12	+5VDC OUT	5V Out				

Input/Output Signal

P type

Pulse	CW/CCW	STEP/DIRECTION
Input Signal Pulse Input	CW/CCW Pulse Maximum Frequency500Kpps Minimum Pulse Width0.8μsec Voltage Level H >+3.0V (+24VMAX. 7mA-15mA) Voltage Level L<+0.8V	Step Pulse Maximum Frequency500Kpps Minimum Pulse Width0.8μsec Voltage Level H >+3.0V (+24VMAX. 7mA-15mA) Voltage Level L<+0.8V

V type - These functions have been included into the C Type motor.

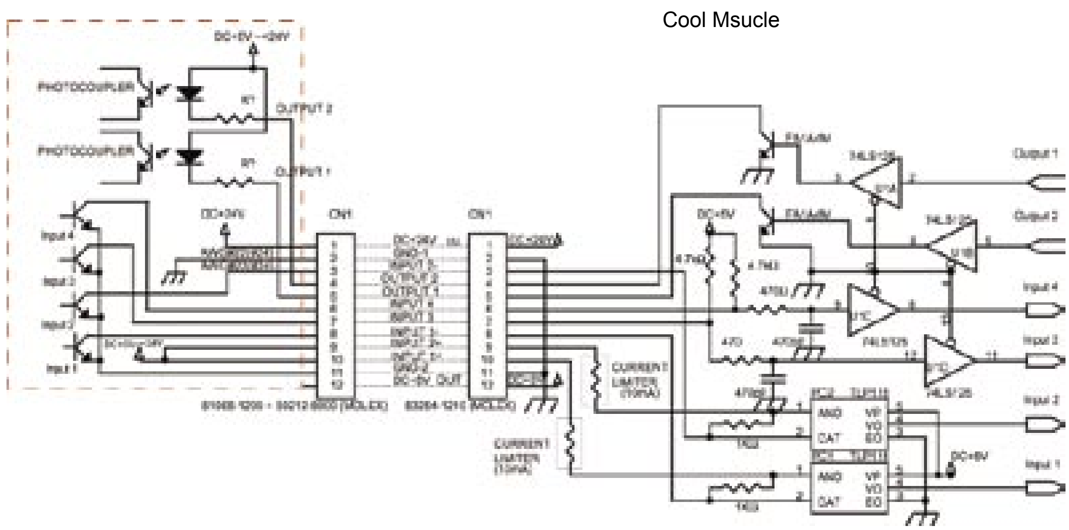
Analog	
Input Signal Variable Voltage Input	Speed Control setting Increase from 2.6 V to 4.8DC, increases speed CW. Decrease from 2.4V to 0VDC, increases speed CCW. Use OP AMP for max. resolution.
	Position Control setting Travel distance is proportinate the the input voltage. Set the max. distance by parameter.

C type

Computer Control Inter face	
Input Signal Control	Via supplied Cable Voltage Level H >+3.0V (+24VMAX. 7mA-15mA), Voltage Level L<+0.8V

Connection Diagram

Controller example



Appendix 2
Parameters

Parameter #	Descriptions	Min.	Max.	Default	Unit	Details	Applicable Types
K20	Baud Rate	0	5	0	K	Sets the Baud Rate 0: 38.4 1: 9.6 2: 19.2 3: 57.6	
K21	Full/Semi Closed	0	36	0	0.1 degrees	Sets the motor to either Full or Semi Closed mode. When set to other than 0, it sets the motor to semi closed mode and the vector angle at which the motor returns to full closed mode. 0: full closed loop 1-36: vector angle	
K22	Time detaly for Semi-closed mode	10	1000	10	msec	Sets the time delay until the motor gotes into an open mode after in-position signal.	
K23	Motor Status	0	1	0		sets the motor to Polling or Event Driven 0: Polling 1: Evemt Driven	
K24	Timed Trigger Interval	10	32767	1000	pulses	Sets the distance interval for the timed trigger.	
K25	Time delay for Slow Response Signal	1111	9999	3333	0.1sec	Sets time delay for slow response signal. Input order is Pin4, Pin3, Pin2, Pin1.	P,V,C
K26	Input logic	0000	1111	0000		Sets Input Logic. 0:Ture in Low 1:Ture in High	P,V,C
K27	Input function at target voltage level (Quick Response Signal)	0000	5555	0000		Assigns a functionat the target voltage level . Input order is Pin4, Pin3, Pin2, Pin1. 0:No Action 1:General Use 2:Origin Sensor 3:Manual Feed CW 4:Manual Feed CCW 5:Output Index signal instead of Inposition Signal	P,V,C

Parameter #	Descriptions	Min.	Max.	Default	Unit	Details	Applicable Types
K28	Input funtcion at the rising edge of Quick Respoinse Signal	0000	9999	0100		Assigns a function at the falling edge of quick response singal. Input order is Pin4, Pin3, Pin2, Pin1. 0:No Action 1:Alarm reset/Pause 2:Motor Free 3:Reset Counter 4:Execute Next Step 5:Execute Previous Step 6:Execute Bank 1 7:Go Origin 8:Manual Jog CW (Execute Bank 2 when K36=2) 9:Manual Jog CCW(Execute Bank 3 when K36=2)	
K29	Input funtcion at the falling edge of Quick Respoinse Signal	0000	9999	0000		Assigns a function at the falling edge of quick response singal. Input order is Pin4, Pin3, Pin2, Pin1. 0:No Action 1:Alarm reset/Pause 2:Enable Motor 3:Reset Counter 4:Execute Next Step 5:Execute Previous Step 6:Execute Bank 1 7:Go Origin 8:Manual Jog CW (Execute Bank 2 when K36=2) 9:Manual Jog CCW(Execute Bank 3 when K36=2)	P,V,C
K30	Input function at target voltage level (Slow Response Signal)	0000	9999	0000		Assigns a function to the target voltage level . Input order is Pin4, Pin3, Pin2, Pin1. 0:No Action 1:General Use 2:Origin Sensor 3:Manual Feed CW 4:Manual Feed CCW 5:Output Index signal instead of Inposition Signal	P,V,C

Parameter #	Descriptions	Min.	Max.	Default	Unit	Details	Applicable Types
K31	Input function at the rising edge of Slow Response Signal	0000	5555	0200		Assigns a function to the rising edge of slow response singal. Input order is Pin4, Pin3, Pin2, Pin1. 0:No Action 1:Alarm reset/Pause 2:Motor Free 3:Reset Counter 4:Execute Next Step 5:Execute Previous Step 6:Execute Bank 1 7:Go Origin 8:Manual Jog CW (Execute Bank 2 when K36=2) 9:Manual Jog CCW(Execute Bank 3 when K36=2)	P,V,C
K32	Input function at the falling edge of Slow Response Signal	0000	5555	0300		Assigns a function to the falling edge of slow response singal. Input order is Pin4, Pin3, Pin2, Pin1. 0:No Action 1:Alarm reset/Pause 2:Enable Motor 3:Reset Counter 4:Execute Next Step 5:Execute Previous Step 6:Execute Bank 1 7:Go Origin 8:Manual Jog CW (Execute Bank 2 when K36=2) 9:Manual Jog CCW(Execute Bank 3 when K36=2)	P,V,C
K33	Output Logic	00	11	00		Sets Output Logic. 0:Normally Open 1:Normally Closed	P,V,C
K34	Output Function	00	66	21		Assigns output function. 0:Command(when using in a command mode you need to set to O) 1:Inposition 2:Alarm 3:General Use1 4:General USs2 5:Analog Output 6:Output Inposition SIignal in Merge Mode.	P,V,C

Parameter #	Descriptions	Min.	Max.	Default	Unit	Details	Applicable Types
K35	Analog Output Function	00	99	31		When Parameter 34 is set to Analog Output (5), choose one of the analog output types. Analog Wave can be monitored in actual size or magnified by 8times. 0: Target position 1: Target position magnified by 8 2:Current Position 3:Current Position magnified by 8 4:Position Error 5:Position Error magnified by 8 6:Current Velocity 7:Current Velocity magnified by 8 8:Current Torque 9:Current Torque magnified by 8	P,V,C
K36	Pulse Interface	0	1	1		Set Pulse type to either CW/CCW or Step/ Direction. 0:CW/CCW 1:Step/Direction C type Cool Muscle 2:enables bank 2 and 3 execution	P
K37	Resolution and Speed unit	00	30	4		Sets resolution and speed unit on the motor. To set the speed unit to100pps, use one of the resorution settings. 0:200, 1:400, 2:500, 3:1000, 4:2000, 5:2500, 6:5000, 7:10000, 8:25000, 9:Reserved , 10:50000 To set the speed unit to10pps, use one of the resorution settings. 20:200, 21:400, 22:500, 23:1000, 24:2000, 25:2500, 26:5000, 27:10000, 28:25000, 29:Reserved , 30: 50000	P,V,C
K38	Analog Interface(V Type)	0	1	0		Sets V type motor to be either Speed Control or Position Control. 0:Speed Control 1:Position Control	V,C

Parameter #	Descriptions	Min.	Max.	Default	Unit	Details	Applicable Types
K39	Voltage Filter Gain	0	1028	128	5[rad/ sec]	Sets cut-off frequency for voltage Low Pass filter.	V
K40	Max. speed for V Type	200	3000	2000	rpm	Sets the max. speed for V type Cool Muscle. Increasing from 2.6V to 4.8V increases speed in a CW direction. At 4.8V the motor reaches the Max. CW speed set by this parameter. Decreasing from 2.4V to 0.2V increases speed. At 0.2V it reaches the max. CCW speed.	V
K41	Travel Range for V Type	-32767	32767	2000	pulses	Sets the Max . travel for V type Cool Muscle. Increasing Voltage from 0.2V to 4.8V changes position in proportion to Input voltage.	V
K42	Go Origin Speed	1	5000	10	100pps	Sets Go Origin speed	P,V,C
K43	Go Origin/Manual feed Acceleration	1	5000	100	$kpps^2$	Sets acceleration for Go Origin and Manual Feed.	P,V,C
K44	Deceration Ratio	10	500	100	%	Sets Deceleration ratio in relation to Acceleration. ie. K44=30 Deceleration is set to 30% of Acceleration.	C
K45	Origin Direction	0	1	1		Sets origin direction 0:CW 1:CCW	P,V,C
K46	Origin Search Method	0	1	0		Sets origin search method 0:Stopper 1:Stopper(start origin search when powered on) 2:Origin Switch 3:Origin Switch (start origin search when powered on)	P,V,C
K47	Origin Stopper Voltage Level	10	100	30	%	When K46 is set to 1 or 2(Origin search by Stopper). you need to set the voltage level at which the motor determins that it has reached the origin. It is set by a percentage of the peak current.	P,V,C

Parameter #	Descriptions	Min.	Max.	Default	Unit	Details	Applicable Types
K48	Offset distance between machine origin and mechanical origin	-32767	32767	0	100 pulses	Sets offset distance between mechanical and electrical origins. When the offset distance is set, it will automatically move to the electrical origin after reaching mechanical origin. When the parameter is set to 0, mechanical and electrical origin are the same. Speed moving from mechanical and electrical origins is the same as go origin speed set by parameter 42.	P,V,C
K49	Manual feed speed	1	5000	10	100pps	Sets manual feed speed.	P,V,C
K50	Manual Jog travel distance	1	100	10	pulses	Sets manual jog travel distance.	P,V,C
K51	Creeping Speed	1	1000	10	100pps	Sets creeping speed	P,V,C
K52	Position P gain	0	300	50	1/256	Sets Position P gain.	P,V,C
K53	Velocity P gain	0	300	250	1/256	Sets Velocity P gain.	P,V,C
K54	Velocity I gain	0	500	2	1/1024	Sets Velocity I gain.	P,V,C
K55	Inposition tolerance	1	100	5	pulses	Sets Inposition tolerance . The motor determines that it has reached the target position, when it gets within the set Inposition tolerance.	P,V,C
K56	Overflow Alarm Level	1	32767	50	K pulses	Set Overflow Alarm Level. When position error is larger than the value set by this parameter, the motor sends out an alarm.	P,V,C
K57	Overload Alarm Time Delay	100	10000	3000	msec	Time delay between overload is detected and alram is sent out.	P,V,C
K58	Software Limit(+)	0	32767	0	100 pulses	Sets limits on the +(CW)side. When set to 0, it ignores software limit.	C
K59	Software Limit(-)	-32767	0	0	100 pulses	Sets limits on the +(CCW)side. When set to 0, it ignores software limit.	C
K60	Push Mode Current Level	10	70	30	%	Sets the current level for Pusch mode. The motor keeps pushing a stopper/wall at the set current for the duraction set by parameter 61.	C
K61	Push time	10	3000	200	msec	Sets the duration which the motor pushes in push mode.	C

see Appendix 9 for information on K64 (Analog Input function)

Appendix 3

Program Mode - Motion Commands

Command	Function	Memory #	Format	Unit	Description	Examples
P	Position	25	PMemory#=Value or PMemory #	pulses	Value holder for absolute positions. Used in CML programs in EEPROM. If P is given a value over 1000000000, the motor spins continously CW . To turn the motor in a CCW set the speed value negative.	P23=2903274 Sets P23 to 2903274 P23 Moves the motor to position 23.
S	Speed	16	SMemory#=Value or SMemory #	100pps	Value holder for speed. Used in CML programs in EEPROM.	S13=20 Sets S13 to 2Kpps S13,P1 Move the motor to P1 at the speed defined by S13.
A	Acceleration	8	AMemory#=Value or AMemory #	$1Kpps^2$	Value holder for Acceleration. Used in CML programs in EEPROM	A13=200 Set A13 to200Kpps2 A13,S1,P1 Move the motor to P1 at speed S1 and the acceleration defined by A13.
T	Timer	8	TMemory#=Value or TMemory #	msec	Set Timer	T1=1000 Set T1 to 1000msec. S1,A1,P1 T1 After reaching P1, the motor waits for the time period defined by T1.
B	Program Bank	16	B Bank#		B# specifies the Bank program #. This signifies the beginning of a program bank with a CML program file.	B1 S1,A1,P1 S2,A2,P3
C	Call Bank		C Program bank#		Calls a specified program bank and executes.	B1 S1,A1,P1 S2,A2,P3 C2(calls B2 and executes)

Command	Function	Memory #	Format	Description	Examples
J	Jump		J Bank#	Jumps and execute the specified program bank.	B1,A1,S1,P1 J2 B2 S2,A2,P3
O	Output High	2	O Output #	Sets the output specified high/on.	S1,P1 O1
F	Output Low	2	F Output#	Sets the output specified low/off.	S1,P1 F1
I	Input	4	I Input#,True, False	The I command monitors a specified input and will execute an action based on the input state.	I1,C3,C2 If input1 is on execute bank3, otherwise execute bank2.
Q	Push Mode	25	Q Position#	The Q command is a replacement for the P command when a push mode movement is required.	B1 A1,S1,Q1
X	Loop		X#	Specify the number of loop for a specified program bank.	B1 X2 A1,S1,P1 Loops B1 twice
Y	P without a wait	25	Y Position#	When networking the motors, use the Y comand instead of P to make motors execute motion individually.	B1 S1.1,P1.1,S1.2,P1.2 P2.1,Y2.2 Motor 2 executes line 2 without waiting for motor 1 to complete motion.
Z	Q without a wait	25	Z Position#	When networking the motors, use the Z comand instead of Q to make motors execute push mode movement individually.	B1 S1.1,P1.1,S1.2,P1.2 P2.1,Q2.2 Motor 2 executes push mode without waiting for motor 1 to complete motion.
T0	No action		I1,C1,T0	Use with the I command. It is a place holder. T0(ZERO)	3,T0,C2 If Input 3 is on the motor does not do anything. If input 3 is of f it executes bank2.
/	Comment		/comment	/ begins a commnt in a CML program file. All comments are stripped when a file is downloaded into a motor.	B1/move to the first hole.

Program Mode - Execution Commands

<table>
<tr><th>Command</th><th>Function</th><th>Format</th><th>Description</th><th>Examples</th></tr>
<tr><td>[</td><td>Execute Bank program</td><td>[bank#</td><td>Sets the output specified high/on.</td><td>[1
Executes bank program1</td></tr>
<tr><td>></td><td>Execute next line</td><td>></td><td>Executes the next line in the program bank.</td><td>>
Executes the first step in bank 1</td></tr>
<tr><td><</td><td>Execute previous line</td><td><</td><td>Executes the previous line in the program bank.</td><td><
Executes the previous line in the current bank program.</td></tr>
<tr><td>|(bar)</td><td>Go Origin</td><td>|(bar)
|1
|2</td><td>|:Make the motor go origin
|1:Make the motor go to position0
|2:Sets the current position to 0</td><td></td></tr>
<tr><td>]</td><td>Immediate Stop/Pause</td><td>]
]]</td><td>Pauses the motor . To resume the program enter [bank#.
Pauses the motor and rests the program bank to its first command line.</td><td></td></tr>
<tr><td>}</td><td>Stop after current motion</td><td>}</td><td>Stop the motor when it cmopletes current motion.</td><td></td></tr>
<tr><td>)</td><td>Motor Free</td><td>)</td><td>Turns the motor into a free state</td><td></td></tr>
<tr><td>(</td><td>Enable Motor</td><td>(</td><td>Enables the motor in a free state.</td><td></td></tr>
</table>

Other commands

Dynamic Mode - Motion Commands

Command	Function	Format	Unit	Description	Examples
P	Position	P= posiiton	pulses	Define the position in dynamic mode.	P=2903274
S	Speed	S=Speed	100pps	Define the speed in dynamic mode.	S=20
A	Acceleration	A=Acceleration	$1Kpps^2$	Define the acceleration in dynamic mode.	A=200

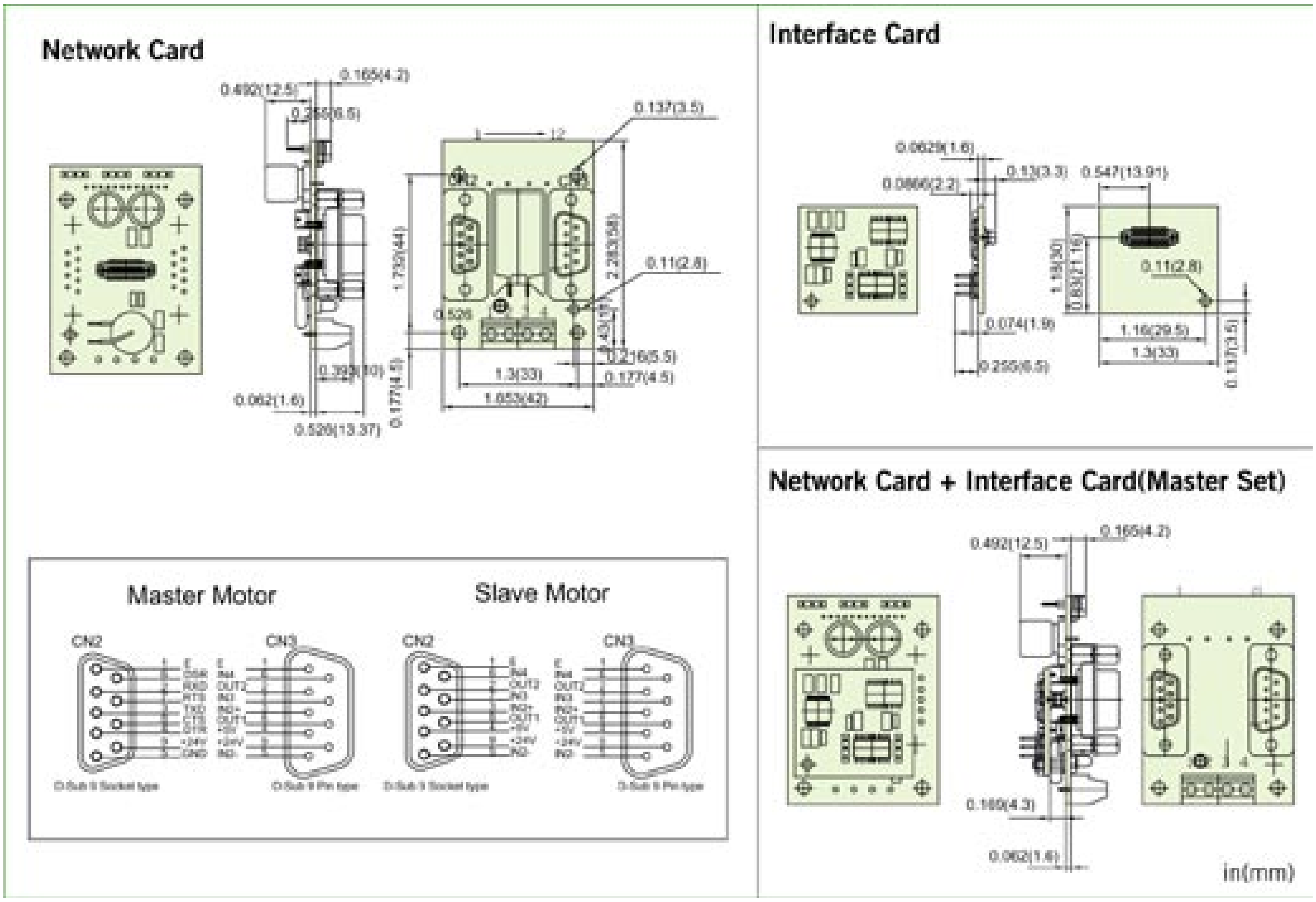
Dynamic Mode - Execution Commands

<table>
<tr><th>Command</th><th>Function</th><th>Format</th><th>Description</th><th>Examples</th></tr>
<tr><td>^</td><td>Execute dynamic commands</td><td>^</td><td>This command exectes dynamic commands in PTP motion.</td><td>P=1000
A=200
S=300
^</td></tr>
<tr><td>~</td><td>Execute CP motion with dynamic commands</td><td>~</td><td>This command executes CP motion.</td><td>P=10
~
P=15
~
P=20
~</td></tr>
<tr><td>|(bar)</td><td>Go Origin</td><td>|(bar)
|1
|2</td><td>|:Make the motor go origin
|1:Make the motor go to position0
|2:Sets the current position to 0</td><td></td></tr>
<tr><td>]</td><td>Immediate Stop/Pause</td><td>]
]]</td><td>Pauses the motor . To resume the program enter [bank#.
Pauses the motor and rests the program bank to its first command line.</td><td></td></tr>
<tr><td>)</td><td>Motor Free</td><td>)</td><td>Turns the motor into a free state</td><td></td></tr>
<tr><td>(</td><td>Enable Motor</td><td>(</td><td>Enables the motor in a free state.</td><td></td></tr>
</table>

Save and Inquiry Commands

Command	Function	Format	Description
+	Relative		Using the + directly after the P # changes the value indicated by P# command into a relative distance in a CW direction. To turn the motor in a CCW direction use a negative value for the speeds.
$	Save		Write to EEPROM
?	Motor Query		?:0~16:Bank # ?70:input status(D0-D7) ?71:input status(D8-D15) for expansion ?73:analog input valueCH1 ?74:analog input valueCH2 ?75: counter ValueCH1 ?75:counter ValueCH2 ?90:all parameters ?91:position list ?92 :speed list ?93: acceleration list ?94: timer list ?95:position error ?96:current position ?97:current speed ?98:current torque ?99:motor status 0: Running 1: overflow 2: overspeed/regenerative voltage 4: overload 8: Inposition

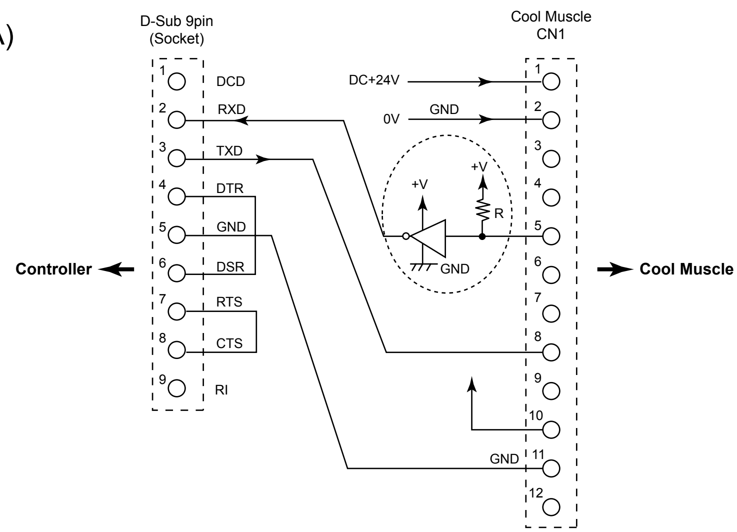
Appendix 4
Network Card Specifications



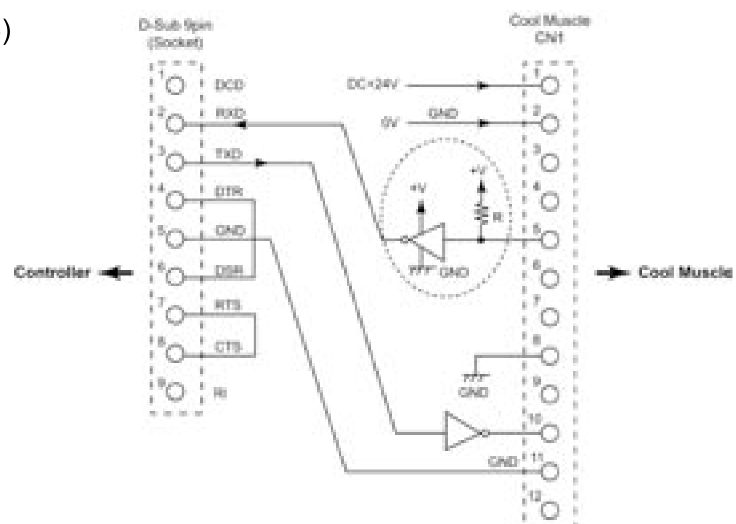
Appendix 5
Custom Interface Cable

The Cool Muscle can be connected to a RS232 port on your controller or PC via the Y cable or the Master Motor set (Network Card + Interface card) that can be purchased from Muscle Corporation. PC RS-232 ports generate -+12V signals which are not directly compatible with the Cool Muscle serial input. You can also make your own cable to connect the Cool Muscle to a controller running TTL level signals as shown below.

A)



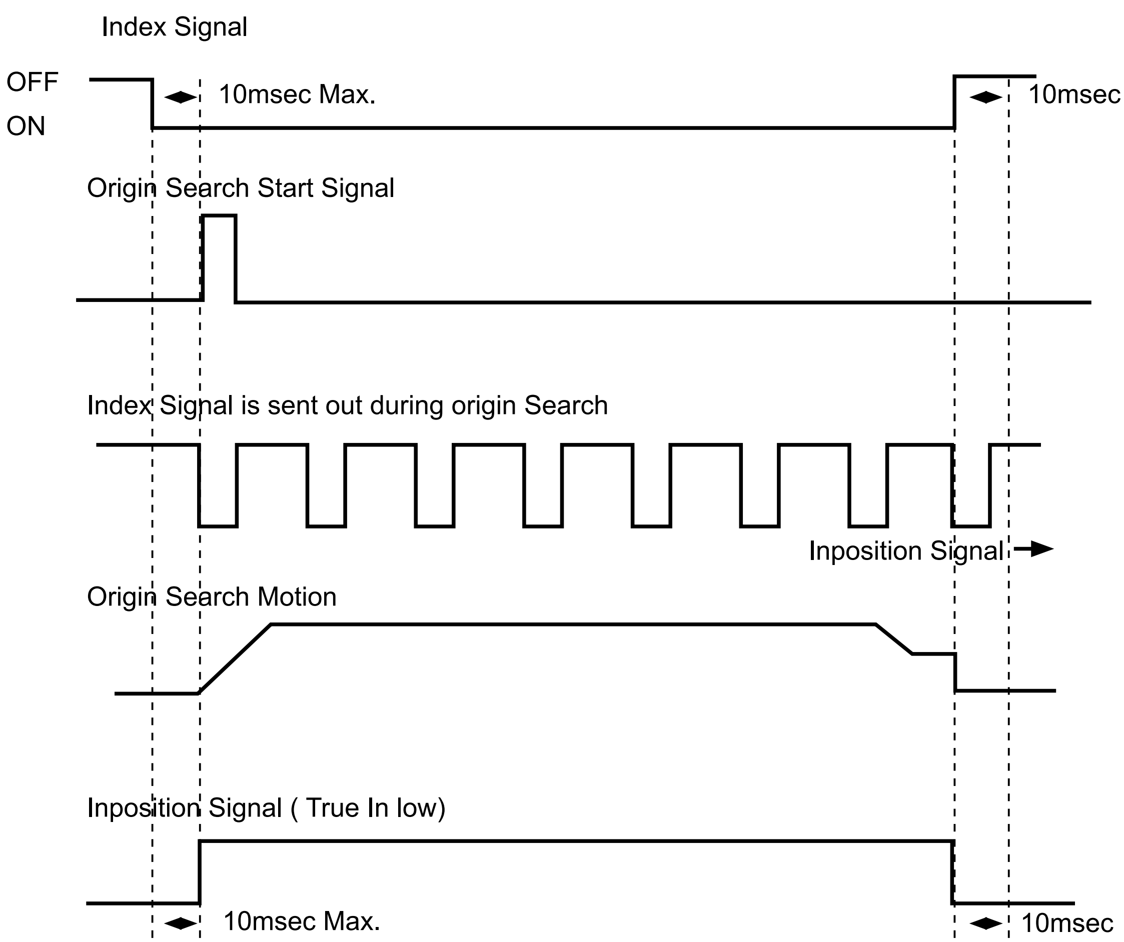
B)



Appendix 6

Index Signal

Instead of an in-posiiton signal you can have the Cool Msucle output index signals during origin search. Set K27 or K30 to 5 and K34 to 1.



Appendix 7

RS-485 Communications

In order to connect a Cool Muscle to a RS-485 serial connection, the motor's parameters need to be adjusted and a CM1DC1-MBS Master Network Card needs to be connected as an interface to the motor. Please take care to set jumpers 1 and 2 on the Master Card's Serial Daughter Card to the appropriate settings. This can be found on page 16 of this manual.

The ID scheme with the RS-485 system is different from that of the RS-232 daisy chain network.
{ is used to preface the motors ID as follows:
{1 - {256 can be used as motor IDs.
{998 addressed all even number ID nodes.
{999 addressed all odd number ID nodes.
{1000 addresses all nodes

A RS-485 motor ID is stored in parameter K62. If K62=0, the motor works as if it is on an RS-232 connection. As RS-485 devices share the communications wires, automatic ID selection is not possible as it is with the daisy chain network. This requires the motor's ID to either be set prior to inclusion into the network or via the D command. The D command allows unaddressed motors to be addressed by using the motor's serial number as the identifying feature. Sending D*id*=serial# across the network will request the motor with the matching serial# to assign itself the *id.*

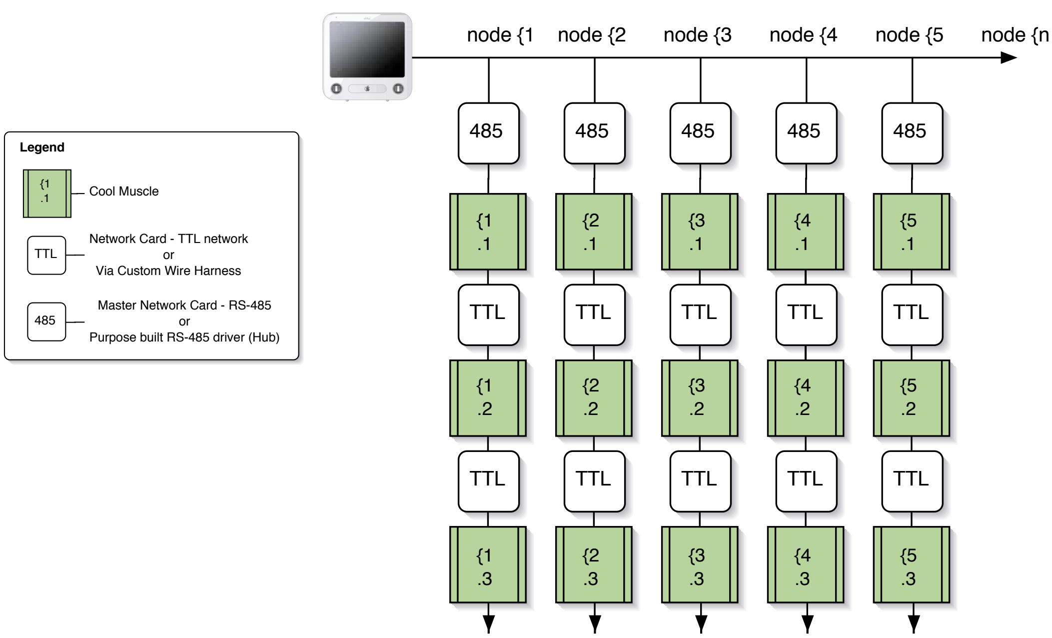
{0 is a software handshaking command that opens the RS-485 communications channel so that a motor can send data. When a motor sends data, the communications channel is automatically closed to other motors. This system prevents communications conflict over the shared wires.

RS-485 Communications Example.

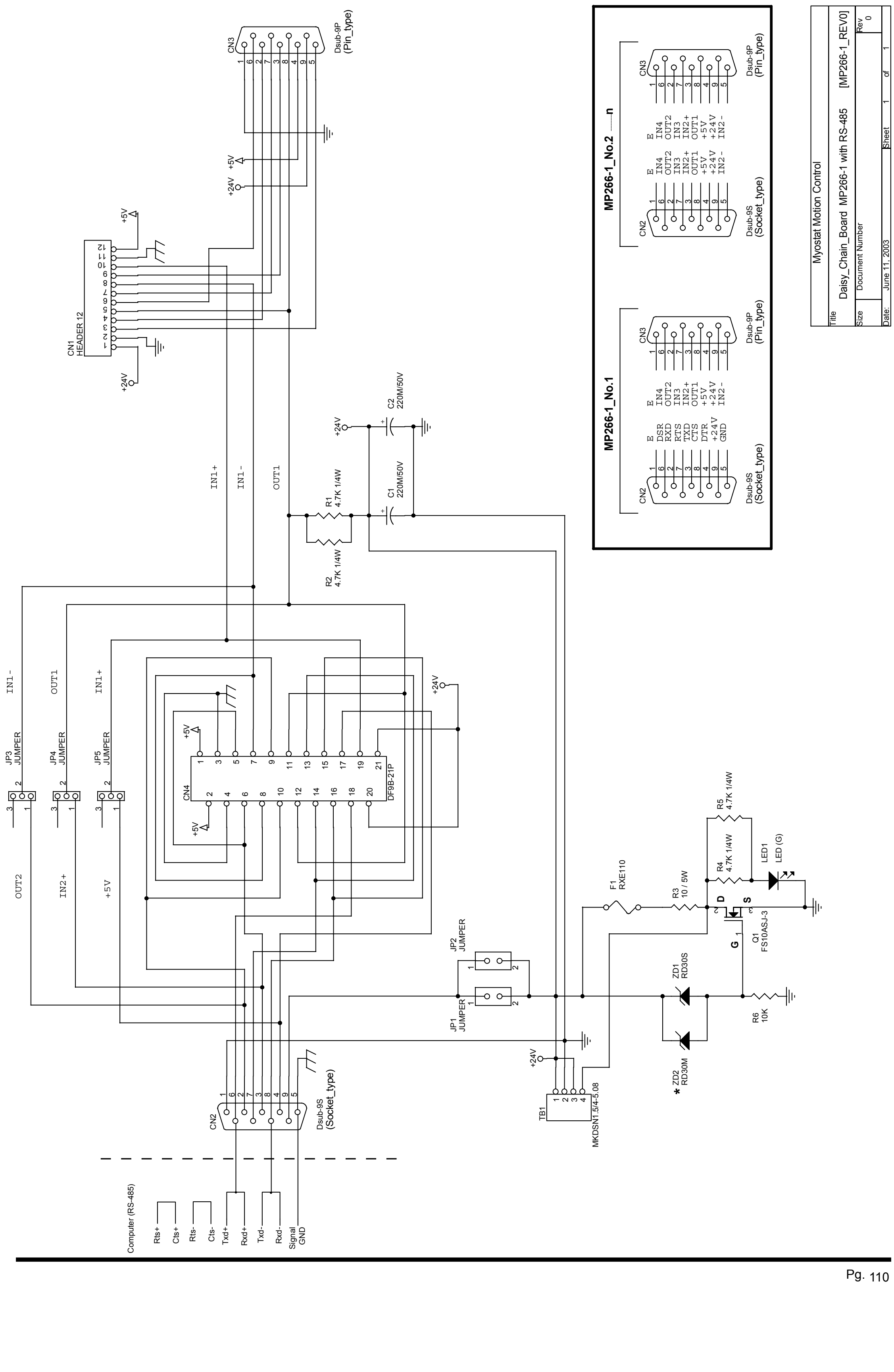
```
{1          (generated by host PC)
[1
{2
[1
{0
{1          (generated by motor 1)
Ux.1=8
{0
{2          (generated by motor 2)
Ux.1=8
{0
```

Possible Combined RS-485/RS-232 Network Layout

The Cool Muscle networking scheme allows for both RS-485 devices and daisy chained devices. Each RS-485 device becomes a node with the RS-485 ID becoming a supernet address for the motors conencted under it.



RS-485 Wiring Schematic (PC to CM1DC1-MBS)



Appendix 8

Using the Analog Inputs with a C-Type Motor

Starting with Firmware Version 1.12, it is possible to use the analog inputs on the C-Type Cool Muscle to vary the motor's operation. All of the analog functions are tired to Input 4 on the motor. K39 is used to set Input 4's software low pass filter. Please refer to page 16 for wiring information. Input 4 can be accessed via the Network Card. Please refer to page 111 for this wiring information.

K64 is the parameter that is used to pick the function of the Analog input:

K64=0: No analog function

K64=1: Input 4 varies the value of S0. Use K40 to set the speed range (+ and -).

K64=2: Input 4 varies the value of P0. Use K41 to set the postion range from 0 to 100xK41.

K64=3: Input 4 varies the value of S13. Use K40 to set the speed range (+ and -).

K64=4: Input 4 varies the value of P24. Use K41 to set the postion range from 0 to 100xK41.

K64=7: Input 4 varies all speeds by muliplying the target speed by a value. The multiplier's value's range is defined by K40. The range is from 0 to K41/100. K41's values can be between 0 and 4000.

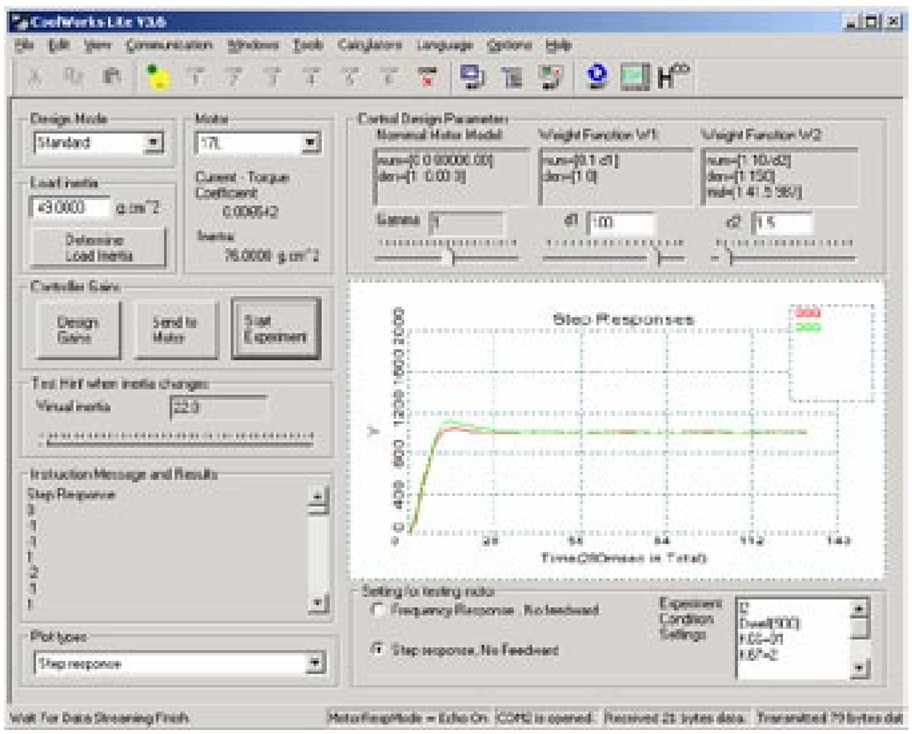
K64=8: Input 4 varies all postions by muliplying the target postion by a value. The multiplier's value's range is defined by K41. The range is from 0 to K41/100. K41's values can be between ±32767.

K64=9: Mimics the V-Type motor settings. When K38=0, Input 4 controls speed with K40 setting the speed range. When K38=1, Input 4 controls position with K41 setting the position range. Please refer to page 55 and 56 for a more detailed description.

Appendix 9

Using the H Infinity Auto Tuning Module (V2 Firmware)

The AutoTuning Hinf designing module within Cool Works Lite V4 is as follows:



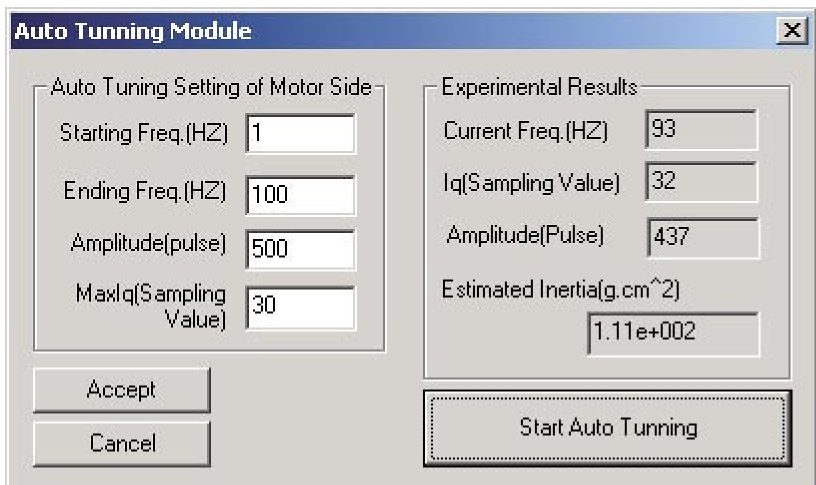
The Main Steps in Designing an H Infinity Controler
Input or estimate the load inertia.
Choose a suitable Response Rate (d1) and Robustness (d2).
Design controller by click "Design Gains" button.
Evaluate the design results through simulation
Send the design results to motor and activate step and frequency response tests.

When you change d1 and d2, the controller gains are recalculated automatically. The intermediate results are shown in the "Instruction Message and Results" window and graphic window.

There are three inertias that will be used in this doucment.
“Load inertia for design” is the inertia value used during Hinf controller design.
“Load inertia for simulation” is inertia value used when displaying a simulation with the designed Hinf controller.
“Estimated inertia” is the inertia of your payload estimated by auto tunning module.

5.1 Design example of a 17L motor

We will begin by designing a Hinf controller for a 17L motor. First, we need to estimate the payload attached to the motor. Click the “Determine Load Inertia” button, you will get the following window:



Using the default setting, click the “Start Auto Tuning” Button. The Auto tuning module starts to work. The auto tuning process continues for 56 seconds as the motor oscilates in frequencies ranging from 1 to 100Hz. The final estimated inertia in this description is 110 gcm^2. You can run the auto tuning routine several times. You should find that all the estimated inertia results are very close.
Remark: In this example, the payload for the motor is a coupling with an inertia is 9.0gcm^2. The inertia of the position sensor is around 30gcm^2. The standard inertia of the 17L rotor is 76gcm^2. The total payload is 9+30+76=115gcm^2.

Clicking the “Accept” button will set the load inertia in Hinf design to 111-76 =34gcm^2.

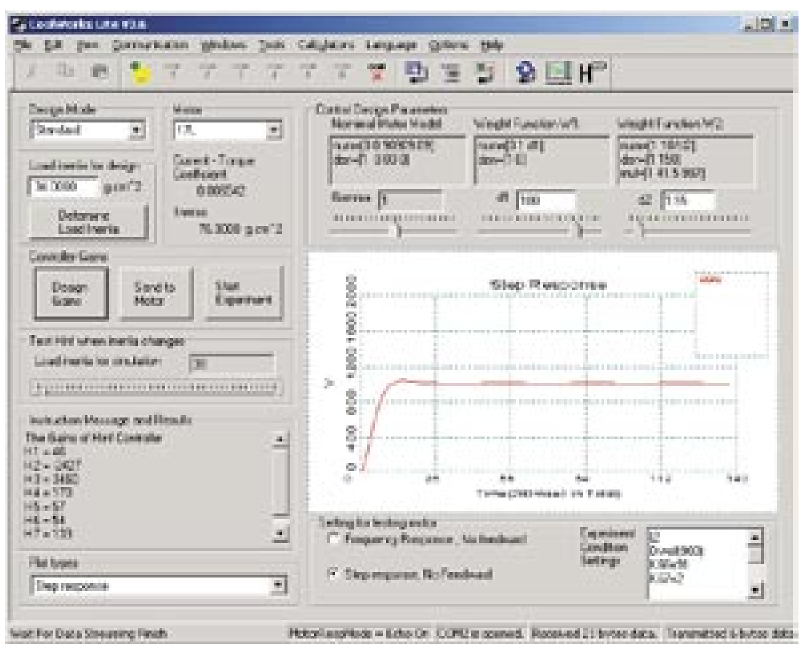
Remark: You can choose different inertia for design. Usually, if you design a controller for heavier load inertias, it works fine if the real load inertia is smaller than that. For the 17L case, we can even set the load inertia for design to 760gcm^2 which is 10 times greater than that of the 17L rotor.

Choose d1 and d2 for controller design

We choose d1=100 & d2=1.55 by moving the sliders on the top right of the window. The resulting gains of the Hinf controller are as follows:

H1 = 46
H2 = -2427
H3 = 3460
H4 = 170
H5 = 57
H6 = 54
H7 = 133

The H0 default is set to 100. The step response of the closed loop system is shown in the “step response” window. It says that the motor works fine.

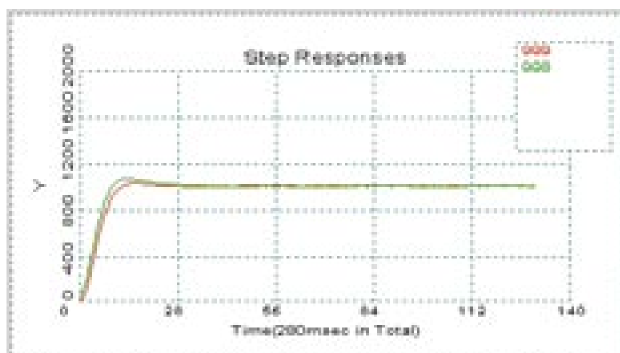


Simulation for a special load

The simulation of the step response is done under the assumption that the real load is equal to the load used in control design. However there are many cases in which the Hinf controller can tolerate inertia uncertainty. Moving the slider labeled “Load inertia for simulation” changes the simulation to show the step response for the designed controller using your chosen simulated inertia. We have set “Load inertia for simulation” to 38gcm^2. The simulated step response is shown as the red trace.

Test and verification - Test of step response

To verify the design result, click the “Send to Motor” button. This sends the designed gains in the “Instruction Message and Results” window to motor. Choose the “Step Response” option in the right bottom corner and click “Start Experiment” button. This will force the motor to move a step with an altitude of 1000 pulse (50000pulse/rev). The result is as follows. Two step responses match very well. The red lineis the simulated one and green one is the real response from motor.

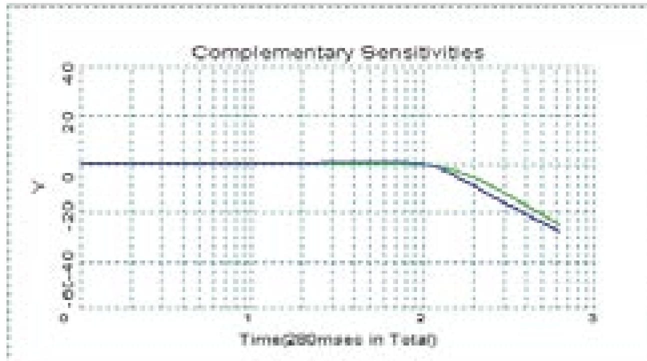


The close trace of the green line, representing the actual move, shows our design is successful. Designing the Hinf controller is that simple!!!

Test of frequency response.

Choose the option “Frequency Response” in the right and bottom corner. To compare of the simulated frequency response to the real one, we must calculate the frequency response one time. This can be done by choosing the “Complementary sensi-

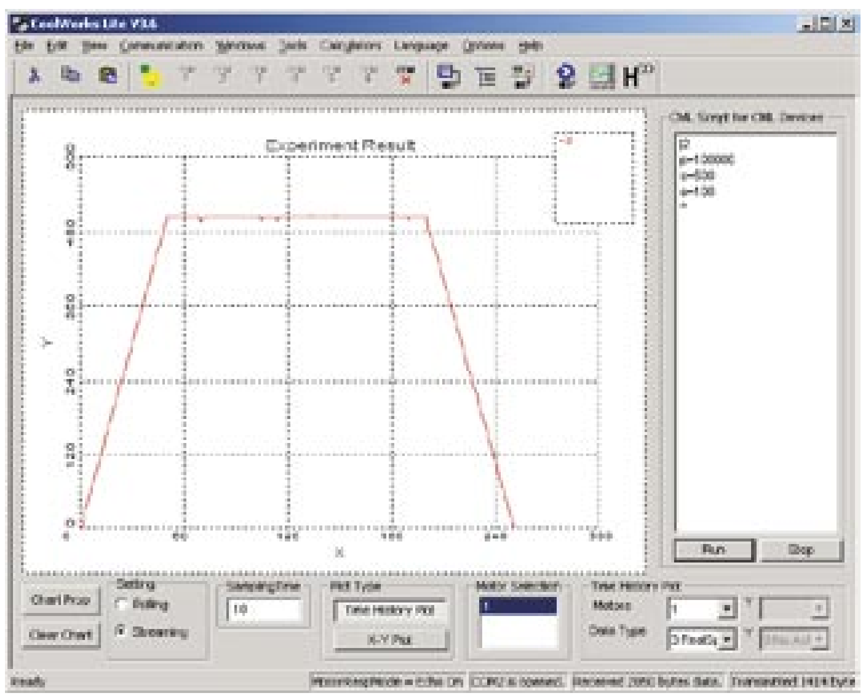
tivity (T)" in the combo box "Plot types" in the left bottom corner. Click on the "Start Experiment" Button. You get the frequency responses as follows.



They match well too.

Test the real motion

Going to Chart window, you can do the following test by setting a motion of p=100000, s=500,a=100,k37=3. *This will rotate the motor 100 times*. Execute the motion by sending ^.



K Parameters

K #	Descriptions	Unit	Details
K20	Baud Rate	K	0: 38.4 1: 9.6 2: 19.2 3: 57.6
K21	Full/Semi Closed	0.1°	0: Full closed loop 1-36: Vector angle
K22	Time delay for Semi-closed mode	msec	Min:10 Max:1000
K23	Motor Status Events		0: Poling Only 1: Running Status, Overflow, Over speed, Over Load/Current, INPOS, circle function and pushing torque 2: Input status 4: Output status 8: Status for Start, Instop, Enable, Diable, Inpause
K24	Timed Trigger Interval	Pulses	Min:10 Max:32767 (active when K34=7)
K25	Time delay for Slow Response Signal	0.1sec	Min:1 Max:9 i.e. K25=3133
K26	Invert Input Signal		0:True 1:False i.e. K26=1101
K27	Input function at target voltage level of Quick Response Signal		0: No Action 1: General Use 2: Origin Sensor 3: Manual Feed CW 4: Manual Feed CCW 5: Output Index signal instead of Inposition Signal 6: Execute Bank 1 **ON STARTUP** 7: Emergency Stop 8: Full Stop
K28	Input function at the rising edge of Quick Response Signal		0:No Action 1:Alarm reset/Pause 2:Motor Free 3:Reset Counter 4:Execute Next Step 5:Execute Previous Step 6:Execute Bank 1 7:Go Origin 8:Manual Jog CW (Execute Bank 2 when K36=2) 9:Manual Jog CCW(Execute Bank 3 when K36=2)
K29	Input function at the falling edge of Quick Response Signal		Same Functions as K28 except 2: Enable Motor
K30	Input function at target voltage level of Slow Response Signal		Same Functions as K27
K31	Input function at the rising edge of Slow Response Signal		Same Functions as K28
K32	Input function at the falling edge of Slow Response Signal		Same Functions as K28 except 2: Enable Motor
K33	Output Logic		0:Normally Open 1:Normally Closed
K34	Output Function		0:Command 1:Inposition 2:Alarm 3:CML O1/F1 4:CML O2/F2 5:Analog Output 6:Output on completion of Origin Search 7:Timed Trigger 8: Motor Free 9: Torque Limit reached - Push Mode Only

K #	Descriptions	Unit	Details
K35	Analog Output Function		0: Target position 1: Target position magnified by 8 2: Current Position 3: Current Position magnified by 8 4: Position Error 5: Position Error magnified by 8 6: Current Velocity/16 7: Current Velocity /2 8:I q Real 9: Iq*8
K36	Pulse Interface		P type 0:CW/CCW 1:Step/Direction C type - 2: enables bank 2 and 3 execution
K37	Resolution and Speed unit	Speed Unit 100pps Speed unit 10pps Speed unit 1pps	0:200, 1:400, 2:500, 3:1000, 4:2000, 5:2500, 6:5000, 7:10000, 8:25000, 10:50000, 40:300, 42:600, 43:800, 44:1200, 45:1500, 46:3000, 47:4000, 48:6000, 49:8000, 50:12000 20:200, 21:400, 22:500, 23:1000, 24:2000, 25:2500, 26:5000, 27:10000, 28:25000,, 30: 50000 60:300, 62:600, 63:800, 64:1200, 65:1500, 66:3000, 67:4000, 68:6000, 69:8000, 70:12000 100: 50000
K38	Analog Interface(V Type)		0:Speed Control 1:Position Control
K39	Voltage Filter Gain	5[rad/sec]	Min:0 Max1028
K40	Max. speed for V Type	rpm	Max speed at 4.8V
K41	Travel Range for V Type	Pulses	Min:-32767 Max32767
K42	Go Origin Speed	100pps	Min:1 Max5000
K43	Go Origin/Manual feed Acceleration	$kpps^2$	Min:1 Max5000
K44	Deceleration Ratio	%	Min:10 Max500
K45	Origin Direction		0:CW 1:CCW
K46	Origin Search Method		0:Stopper 1:Stopper(start origin search when powered on) 2:Origin Switch 3:Origin Switch (start search when powered on)
K47	Origin Stopper Voltage Level	%	Min:10 Max:100
K48	Offset distance between machine origin and mechanical origin	100 pulses	Min:-32767 Max:32767
K49	Manual feed speed	100pps	Min:1 Max:5000

K #	Descriptions	Unit	Details
K50	Manual Jog travel distance	Pulses	Min:1 Max:100
K51	Creeping Speed	100pps	Min:1 Max:1000
K55	Inposition tolerance	Pulses	Min:1 Max:1000
K56	Overflow Alarm Level	K pulses	Min:1 Max:32767
K57	Overload Alarm Time Delay	msec	Min:100 Max:10000
K58	Software Limit(+)	100 pulses	Min:0 Max:32767
K59	Software Limit(-)	100 pulses	Min:-32767 Max:0
K60	Push Mode Current Level	%	Min:10 Max:80
K61	Push time	50msec	Min: 1 Max:30001
K62	RS-485 Node ID		0: RS-232 mode 1~256: RS-485 ID -1~-256: RS-485 Node ID, no auto report function
K63	External Encoder Input		0:None 1:Phase A only 2:Phase A and B
K64	Analog Inut Function		0:None, 1:S0, 2:P0, 3:S13, 4: P24 5:S14, 6:P25, 7:Speed 0-Set speed, 8:Position Mulitplier 9:Analog control only, K38
K65	Slave Motor Baud Rate		0:38.4, 1:9.6, 2:19.2, 3:57.6, 4:76.8, 5:129, 6:173, 7:515 ***Set baud rate starting with the last motor on the network.***
K66	Data Streaming		0: None. 1: Send back speed target 2: Send back real position 3: Send back real speed 4: Send back real current Iq 5: Position Real 6: Velocity Real 80: Torque data streaming (cNm)
K67	Data Streaming Sample Timing	ms	0 - 30000
K68	S Curve Function		0:S Curve with fixed timing 1:S Curve without timing
K69	S Curve Gain		0 - 1024